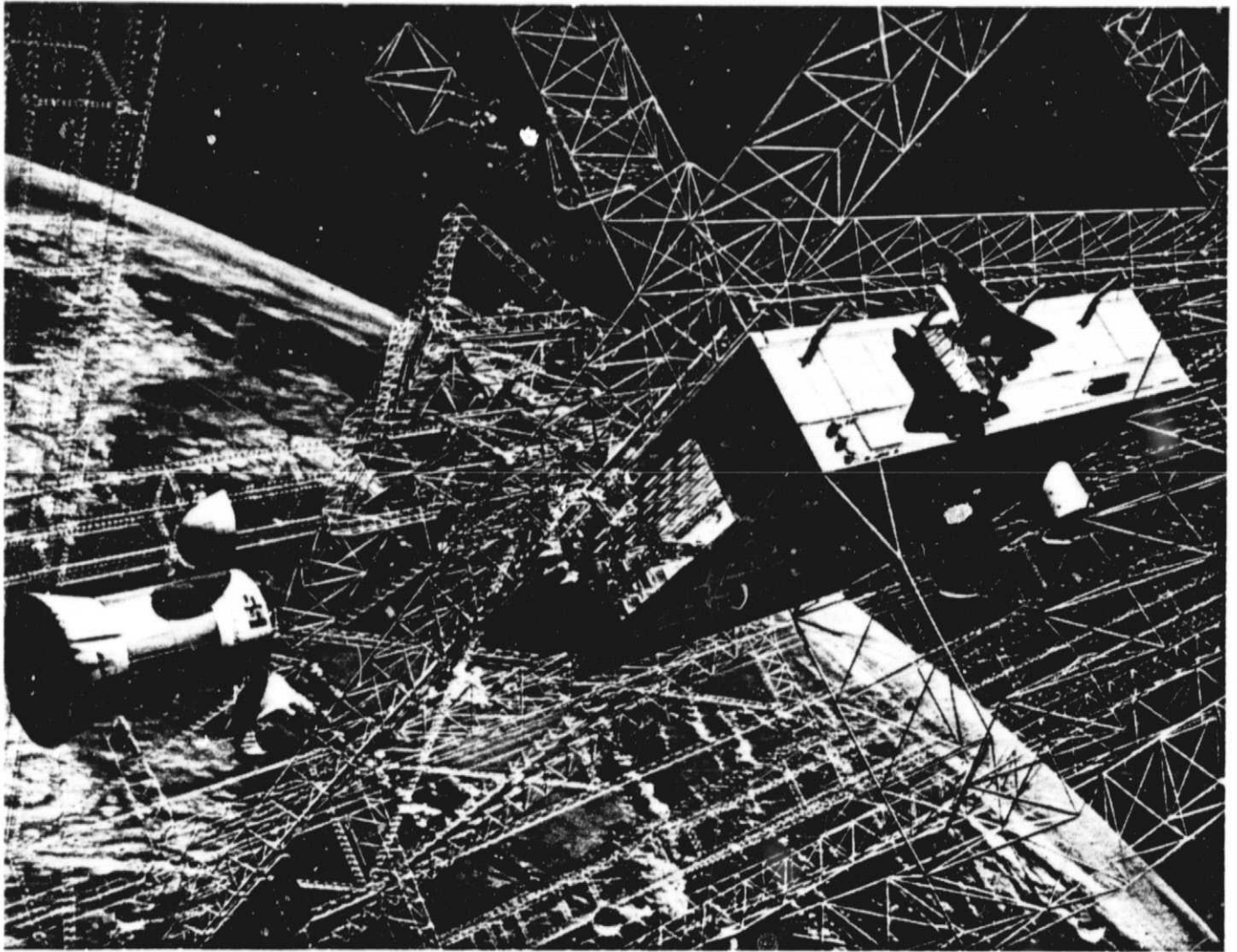


## **General Disclaimer**

### **One or more of the Following Statements may affect this Document**

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

# Study of Robotics Systems Applications to the Space Station Program



Submitted to:

Innovative Utilization of the  
Space Station Program, Code MFA-13  
NASA Headquarters  
Washington, DC 20546



**kms  
fusion**  
inc.

Ann Arbor, Michigan

# Study of Robotics Systems Applications to the Space Station Program

Report Number KTR-108

October, 1983

Ann Arbor, Michigan

Submitted to:

Innovative Utilization of the  
Space Station Program, Code MFA-13  
NASA Headquarters  
Washington, DC 20546  
for Contract NAS W-3751

Submitted by:

James C. Fox, Principal Investigator  
KMS Fusion, Inc., Ann Arbor, Michigan  
and the  
Coinvestigative Staff at the  
Center for Robotics and Integrated  
Manufacturing,  
University of Michigan

**kms**  
**fusion**  
inc.  
Ann Arbor, Michigan

-INDEX-

FINAL REPORT ON THE  
STUDY OF ROBOTICS SYSTEMS APPLICATIONS TO THE  
SPACE STATION PROGRAM

|   | Page |
|---|------|
| List of Figures & Tables                                | v    |
| Acknowledgement   | vi   |
| Foreword  | vii  |
| 1. Introduction   | 1    |
| 1.1 Background and Scope                                | 1    |
| 1.2 Objectives and Approach                             | 1    |
| 1.3 Executive Summary                                   | 2    |
| 2. Typical Mission Definition                           | 4    |
| 2.1 Robotics Application Guidelines and Assumptions     | 4    |
| 2.1.1 Automated Operation Benefits                      | 4    |
| 2.1.2 Robotic Application Scenarios                     | 4    |
| 2.1.3 System Implementation Assumptions                 | 5    |
| 2.2 Task Environment and Facilities                     | 5    |
| 2.2.1 Construction Platform Features                    | 5    |
| 2.2.2 Supervisory Station Features                      | 6    |
| 2.3 Robotic Subsystem Functions                         | 6    |
| 2.4 Support Subsystem Characteristics                   | 7    |
| 2.4.1 Guidance and Control Subsystem                    | 7    |
| 2.4.2 Communications Subsystem                          | 8    |
| 2.4.3 Energy Transmission and Storage                   | 9    |
| 2.4.4 Latching Subsystem                                | 9    |
| 3. Robotic System Requirements and Technology Forecasts | 11   |
| 3.1 System Requirements and Features                    | 11   |
| 3.1.1 Overall Functional Description                    | 11   |
| 3.1.1.1 System Control and Coordination                 | 11   |
| 3.1.1.2 Operational Assumptions                         | 13   |
| 3.1.1.3 Typical Assembler Task Scenario                 | 14   |
| 3.1.2 Robot Structure and Control                       | 16   |
| 3.1.2.1 Manipulator Arms                                | 16   |
| 3.1.2.2 Grippers and End Effectors                      | 17   |
| 3.1.2.3 Control Systems                                 | 22   |
| 3.1.3 Sensing and Machine Vision                        | 22   |
| 3.1.3.1 Vision Sensors                                  | 24   |
| 3.1.3.2 Force Sensors                                   | 24   |
| 3.1.3.3 Tactile Sensors                                 | 25   |
| 3.1.4 Computers, Languages and Communications           | 25   |
| 3.1.4.1 Requirements for Space Computers                | 25   |
| 3.1.4.2 Robot Languages                                 | 27   |
| 3.1.4.3 Communication Requirements                      | 28   |
| 3.1.5 Expert and Knowledge Systems                      | 29   |
| 3.1.5.1 Problem Solving                                 | 29   |
| 3.1.5.1.1 Distributed Problem Solving                   | 29   |

|   |    |
|---|----|
| 3.1.5.1.2 Spatial Reasoning and Motion Planning           | 30 |
| 3.1.5.2 Knowledge Based Systems                           | 30 |
| 3.1.5.2.1 Assembly Activities                             | 30 |
| 3.1.5.2.2 Inspection and Technician Activities            | 31 |
| 3.2 Current Capability Analyses and Research Directions   | 32 |
| 3.2.1 Robot Structures and Control                        | 32 |
| 3.2.1.1 Kinematics Analysis                               | 32 |
| 3.2.1.2 Robot Arm Dynamics                                | 35 |
| 3.2.1.3 Robot Motion Control Methods                      | 37 |
| 3.2.1.4 Robot Control Research Directions                 | 44 |
| 3.2.2 Sensing and Machine Vision for Space Robots         | 45 |
| 3.2.2.1 Contact Sensors                                   | 46 |
| 3.2.2.2 Noncontact Sensors                                | 47 |
| 3.2.2.3 Machine Vision                                    | 48 |
| 3.2.2.3.1 Binary Vision and Gray Scale Vision             | 48 |
| 3.2.2.3.2 Machine Vision Subtask Areas                    | 49 |
| 3.2.3 Robot Languages, Computers, and Data Communications | 51 |
| 3.2.3.1 Robot Languages                                   | 51 |
| 3.2.3.2 Computers   | 53 |
| 3.2.3.3 Computer Communication Networks                   | 54 |
| 3.2.4 Expert and Knowledge Based Systems                  | 57 |
| 3.2.4.1 Problem Solving                                   | 57 |
| 3.2.4.2 Knowledge Based and Expert Systems                | 58 |
| 3.3 Technology Forecast                                   | 59 |
| 3.3.1 Generation Dynamics                                 | 60 |
| 3.3.1.1 Computer Generation Dynamics                      | 60 |
| 3.3.1.2 Industrial Robot Generation Dynamics              | 61 |
| 3.3.1.3 Robot Intelligence Generation Dynamics            | 61 |
| 3.3.2 Precursors  | 63 |
| 3.3.3 Robot Structure and Control Technology Forecast     | 64 |
| 3.3.3.1 Overall Structure and Control                     | 64 |
| 3.3.3.2 Electric Motors                                   | 66 |
| 3.3.3.3 End Effectors and Manipulators                    | 67 |
| 3.3.4 Technology Forecast for Sensors                     | 68 |
| 3.3.4.1 Robot Vision Functions                            | 68 |
| 3.3.4.2 Perspective on Visual Sensors                     | 70 |
| 3.3.4.3 Laser Sensors                                     | 71 |
| 3.3.4.4 Tactile Sensors                                   | 71 |
| 3.3.4.5 Force Sensors                                     | 72 |
| 3.3.5 Technology Forecast for Computers and Peripherals   | 72 |
| 3.3.5.1 Robot Computer Architecture                       | 72 |
| 3.3.5.2 Memory Technology                                 | 73 |
| 3.3.6 Technology Forecast for Robotic Software            | 73 |
| 3.3.6.1 Sensor Signal Processing                          | 73 |
| 3.3.6.2 Robotic Language Structures                       | 73 |
| 3.3.6.3 Robotic Artificial Intelligence                   | 76 |
| 4. Support Subsystem Requirements                         | 77 |
| 4.1 Guidance and Control Subsystem                        | 77 |
| 4.2 Communication Subsystem                               | 79 |
| 4.3 Energy Transfer and Storage Subsystem                 | 79 |

|   |     |
|---|-----|
| 4.4 Mechanical Latching Subsystem   | 81  |
| 5. Environmental Requirements   | 83  |
| 6. Continuing Research Area Recommended for<br>Space Applications           | 84  |
| 6.1 Robot Structures and Control  | 84  |
| 6.1.1 Improved Manipulator Design and Control                               | 84  |
| 6.1.2 Integrated Control Structure with Sensory<br>Feedback Information     | 84  |
| 6.1.3 Advanced Control Strategies Utilizing Sensory<br>Feedback Information | 85  |
| 6.1.4 Advanced Sensor Research and Development                              | 86  |
| 6.2 Sensing and Machine Vision  | 87  |
| 6.2.1 Sensors   | 87  |
| 6.2.2 Machine Vision  | 88  |
| 6.3 Computers, Languages and Communications                                 | 89  |
| 6.3.1 Languages   | 89  |
| 6.3.1.1 Hierarchical View of Robot Process                                  | 89  |
| 6.3.1.2 Design of Language Features   | 92  |
| 6.3.1.3 RL Related Subjects   | 93  |
| 6.3.1.4 Analytical Tools for Evaluating RLs                                 | 95  |
| 6.3.2 Space Robot Computers   | 95  |
| 6.3.2.1 Introduction to the Research Problem                                | 95  |
| 6.3.2.2 Architecture of Space Robot Computers                               | 96  |
| 6.3.2.3 Reliability of Space Robot Computers                                | 97  |
| 6.3.2.4 Performance Measures for Space Robot<br>Computers                   | 99  |
| 6.3.3 Research Needs for Space Data Communications                          | 100 |
| 6.4 Distributed Problem Solving and Knowledge<br>Based Systems              | 101 |
| 7. Results and Conclusions  | 103 |
| 8. References   | 105 |
| Appendix I. List of Interviewees for Technology<br>Forecast                 | 111 |

## List of Figures & Tables

| <u>Figure No.</u> | <u>Title</u>                         | <u>Page No.</u> |
|-------------------|--------------------------------------|-----------------|
| 3.1.1-1           | Space Station Robotic System Concept | 12              |
| 3.1.2-1           | Cartesian Coordinate Robot           | 18              |
| 3.1.2-2           | Cylindrical Coordinate Robot         | 19              |
| 3.1.2-3           | Spherical Coordinate Robot           | 20              |
| 3.1.2-4           | Revolute Coordinate Robot            | 21              |
| 3.2.1-1           | Link Coordinate System               | 34              |
| 3.3.4-1           | Robotics Generation Dynamics         | 69              |
| 6.3.1-1           | Hierarchical Structure               | 91              |

| <u>Table No.</u> | <u>Title</u>   | <u>Page No.</u> |
|------------------|--|-----------------|
| 3.2.1-1          | Comparison of Robot Arm Dynamics<br>Computational Complexities | 38              |
| 3.2.1-2          | Comparisons of the PD and Adaptive<br>Controllers              | 43              |
| 3.3.5.2          | Memory Technology Trends                                       | 74              |

### Acknowledgements

The initial concept of studying robotics applications to selected Space Station activities was suggested by Dr. Alex Glass of KMS Fusion. The robotics subsystems material was provided by the Coinvestigator Staff from the Center for Robotics and Integrated Manufacturing at the University of Michigan which included Professors Edward Delp, Ramesh Jain, C. George Lee, Kang Shin and Richard Volz. The technology forecast was provided by Professor Kan Chen from the Program in Technology Assessment at the University of Michigan.

Additional study material was provided by the KMS Fusion Aerospace staff including James Maszatics, Stephen Smith and Douglas Thomas. A special acknowledgement is made to Shirley Hudgins for her expert manuscript preparation services.



## Foreword

The Study of Robotics Systems Applications to the Space Station Program as reported herein has utilized the following definitions for the terms "robot" and "robotics":

ROBOT: A robot is a reprogrammable multi-functional manipulator designed to move material, parts, tools, or specialized devices, through variable programmed motions for the performance of a variety of tasks.

Adopted by the Robot Institute of America

ROBOTICS: Robotics is the study of the basic organization and operation of intelligent computer-based robots. In general, robotics research involves both basic and applied research in

- o Manipulators and Control Structures
- o Sensors
- o Programming Languages
- o Instrumentation and System Architecture
- o Knowledge Based Systems

# Final Report on the Study of Robotics Systems Applications to the Space Station Program

## 1. Introduction

This final report is submitted to NASA Headquarters, Code MFA-13 by KMS Fusion, Inc. in accordance with the study documentation requirement as directed by Contract NASW-3751. The Study of Robotics Systems Applications to the Space Station Program as reported herein represents the results of KMS Fusion's participation in NASA's Innovative Utilization of the Space Station Program.

### 1.1 Background and Scope

In response to the Innovative Utilization Program solicitation issued by the NASA Office of Space Science and Applications, KMS Fusion, Inc. (KMSF) proposed to study the application of semi-autonomous (supervised) robots to perform several types of missions included within the multiple Space Station functions defined by NASA. Although a large number of multi-purpose facility functions have been defined, this report considers robotics systems applications to potential uses of the Space Station as an Assembly Facility and secondarily as a Servicing Facility.

The rationale for investigating the use of robotics systems on-board the Space Station and on other space missions has been driven in part by an expected set of requirements to (1) build large structures in space, (2) maintain and service other platforms, satellites and spacecraft, (3) service manufacturing facilities and (4) provide support for transportation node activities. In general, many of these tasks would require significant and perhaps unachievable levels of EVA man-hours under very hazardous and difficult conditions. Accordingly, a large amount of analysis and planning has been expended by the Government and its contractors in developing the detail rationale for robots in space, for example as reported by R. Korf in the Space Robotics Feasibility Study [Kor82] and by a number of supporting papers presented by the NASA Space Station Technology Working Group members at the Space Station Technology Workshop in Williamsburg, VA in March 1983. Because such a large body of literature exists detailing the rationale for space robotics, this study emphasizes selected robotics applications to the Space Station and refers to the literature as appropriate.

### 1.2 Objectives and Approach

The major objectives of the Space Robotics Study were to (1) define selected typical missions for robotics applications to the Space Station, (2) identify candidate robotic systems functions and characteristics which could be utilized to carry out these missions, (3) evaluate the current technology status of the candidate systems, (4) perform a 1990's technology forecast for the relevant robotic systems and subsystems and (5) identify key research areas to assist in providing for the availability of robotic systems for use in the Space Station operational era. As

part of the technology forecast, expected improvements in the cost vs. performance trade study results were investigated in each of the robotic subsystem areas comprising the candidate systems.

The study approach to achieve the stated objectives was to utilize the space instrumentation systems experience of the KMS Fusion Principal Investigator's staff in collaboration with the robotic systems analysis and research capabilities of the Co-Investigator staff at the University of Michigan's Robot Systems Division in the Center for Robotics and Integrated Manufacturing (CRIM). The initial task consisted of analyzing potential Space Station requirements, defining candidate robotic missions and identifying generic classes of applicable robotic functions to be implemented within the candidate systems. The next set of tasks accomplished were to describe and evaluate the current technology capability level of the selected robotics functions (subsystems) and to perform a technology forecast for the predicted evolution of the cited robotics subsystems to the Space Station era capability level. The last step in the approach was to analyze the predicted requirements vs. capabilities to identify key areas of recommended research which could enhance and improve the outlook for practical implementation of supervised robotic systems applications to the Space Station.

### 1.3 Executive Summary

A typical robotics system mission is described in Section 2 along with the pertinent application guidelines and Space Station environmental assumptions utilized in developing the robotic task scenarios. To conform with the mission requirements, specific functions are defined for the robotic subsystems and the support subsystems. In Section 3, we first provide a functional description of a supervised dual robot space structure construction system and define four key robotic technology areas that will be required to implement the Space Station robotic Assembler. The remainder of Section 3 is devoted to a description of the current status and evolutionary trends of each of the four robotic technology areas and is concluded with a specific technology forecast for each area related to the 1990's time frame.

In Section 4 we present a brief discussion suggesting alternate and candidate technologies for implementing the more routine space technology support subsystems that will be required to support the Space Station robotic systems in assembly and servicing tasks. The support subsystems considered provide the functions of guidance and control, communications, energy storage and mechanical motion and latching. A discussion of the environmental conditions impacting the robotic configuration design and operation is provided in Section 5.

Addressing each key robotic subsystem area, Section 6 presents the rationale for and a brief outline detailing recommended research areas for future study and investigation. In Section 7, we present a synopsis of the study project results and reach several key conclusions including the very strong probability that selected Space Station tasks will ultimately be performed by some

form of supervised robots and that this utilization will be substantiated by future benefits vs cost trade studies.

## 2. Typical Mission Definition

### 2.1 Robotics Application Guidelines and Assumptions

#### 2.1.1 Automated Operation Benefits

In order to carry out space structure assembly and servicing facility tasks as defined for the Space Station, automated operation has been analyzed in the NASA literature and proposed as an efficient approach which promises to provide improved system performance, increased crew safety, higher crew productivity and decreased system life cycle costs. The types of tasks initially under consideration for robotics applications to the Space Station will include relatively simple, repetitive operations which could be performed by one or more robots with some degree of human supervision either in an IVA or in an EVA environment. The benefits to be derived will include the potential reduction both in crew size and in the associated facilities and ground support functions, increased operating periods available from the automated equipment (more operations per day per payload pound) and reduced crew preparation for and exposure to hazardous EVA environments and activities.

#### 2.1.2 Robotic Application Scenarios

The multiple uses of a Space Station base and its associated elements have been recently updated [Hod83] to include functional utilization as (1) an on-orbit laboratory and permanent observatory, (2) a transportation node and staging base, (3) a servicing facility for other elements (e.g. platforms, satellites, etc.), (4) an assembly facility, (5) a manufacturing facility and storage depot and (6) a data management node. In each area of functional operations, specific applications of automated operations could be developed (e.g. "docking" robots could be used to assist in maneuvering and latching tasks during docking operations associated with the transportation node functions) however this report considers mainly the robotic and support subsystems required to perform selected tasks associated with the assembly and servicing facility concepts.

One scenario considered here is the use of a supervised dual robot system which could be stationed on a Space Station construction module for the purpose of performing selected routine tasks in assembling portions of large space structures which in general either exceed the Shuttle cargo bay dimensions and/or are more cost-effective to construct on-orbit. Examples of these types of structures include trussed beams, large solar arrays, antennas, pre-fabricated module section assemblies and the like.

An additional use of the construction or assembler robot system is considered here as resulting from a programmable/commandable capability to also perform servicing facility tasks such as may be encountered in satellite servicing (e.g. deployment, retrieval assist, inspection, test and modular replacement). In this "technician" case the robot system would need to perform more complex functions than in the "assembler" case and hence either

increased artificial intelligence or higher human supervision levels or both would be required.

### 2.1.3 System Implementation Assumptions

Although robot systems are proposed in part to offload the crew activities from selected tasks, it is generally recognized that the primary utilization of robot systems in the 1990's era will be for performing simple tasks which do not require complex intelligence and which can be performed with the assistance of an "on-call" human supervisor. As the complexity and possible number of alternative actions increase for proposed robotic tasks, a commensurate increase must also be provided in the form of expert and knowledge based systems capability leading to an exponential increase in the data storage and computational capacity of the robot system. Accordingly, complex systems requiring increased computer and memory capacity must be traded off against the potential savings in reduced crew utilization. This study assumes an on-going evolution in artificial intelligence, robot based computers, robot software, machine vision and other fundamental capabilities which will result in different tradeoff results of the semi-autonomous level vs human intervention requirement comparisons at different eras in time. In general, the capability to provide higher levels of robot systems autonomy will increase in the future for the same cost of weight and power while maintaining constant or improved performance. At the same time, the study assumes that human intervention will decrease as higher levels of robot intelligence become available.

Additional assumptions utilized in this study include the premise that (1) the Space Station modules supporting the robotic systems will be designed for compatibility with the robotic requirements (i.e. expensive retrofit will be avoided) and (2) the support subsystems providing mobility, communications, energy transmission and storage, and inertial loads transfer do not present significant technological barriers and that suitable candidates and designs will be available for implementation in the Space Station operational era.

## 2.2 Task Environment and Facilities

### 2.2.1 Construction Platform Features

A construction platform or module is assumed to be that Space Station element which provides for the basing of the robotic assembler subsystem and the structural materials to be assembled. This module would afford suitably dimensioned work areas constructed with robot tracks to provide for optimum robot access to the workpieces under assembly and for robot latching to the base. It is also assumed that the construction module structural attachment to the base would provide for relatively easy crew access by EVA to the worksite for maintenance and other servicing activities. The module may be configured in several large bays with several openings for transport or advance of completed structures and with fixed or moveable sunshields for thermal and lighting control. Since visible sensors are assumed, workpiece and

workplace illumination conditions will be important and some systems will require sunshading to protect the sensors and to ensure proper operation of the system. Aside from the sunshading aspects, the robotic and support subsystems will operate in an assumed unprotected deep space environment on the construction platform.

In addition to the mechanical provisions for the robotic systems tracks, it is also assumed that the platform will afford raw materials and unassembled parts storage capacity for system utilization and that raw part feed and completed section handling and stowage may be provided by other automated operations which are not considered in detail in this report. The platform construction will also support the devices (e.g. data links, cables, etc.) that may be required for communication between the robot systems and the supervisor and for energy transmission to the robot subsystems.

### 2.2.2 Supervisory Station Features

The system supervisor's station is assumed as a separate local monitoring station to be implemented within an attached manned element which is part of the Space Station base. The supervisor, working in a controlled shirtsleeve environment, will be provided with (1) a telepresence TV display of each robot activity, (2) a central control panel for commanding and statusing the robots and (3) command lines to the proximate central processor which controls and directs the robotic local processor control systems.

Given this configuration, the human supervisor will have the capability to monitor the robot system activity and to intervene actively when malfunctions appear imminent or are predicted by the system. Additionally, the supervisor can also operate in a "hands-off" mode with a provision to be summoned by assistance requests which may be activated by the robot sensor signals, by a local processor branch to a "no-decision" point, by other proximate sensors in the facility, or by any combination of the above. Assumption of these system capabilities will lead to early and direct tradeoffs in the desired level of system autonomy (on-board expert systems, computing and memory) vs. the required level of human intervention and support needed to carry out the assigned tasks.

### 2.3 Robotic Subsystems Functions

The primary Assembler configuration considered in this report is assumed to be a supervised sensor-based dual robot system where each robot contains its own local processors but is directed by a central processor. For the assembly tasks, the dual robots are assumed to be latched to the base when handling and assembling materials. A sequence of activity commands will be directed by the central processor to the individual robots which will initiate local controlled processes such as carrying out specific manipulator trajectories, gripping work pieces, transferring work pieces and performing all the detail steps required to align and fasten the work pieces. The local and central processors will

provide closed loop control by utilizing contact and non-contact sensors to feedback measurements of range, size, orientation, force, torque, position, rate and other relevant data. At all times the system is subject to human intervention by the supervisor who monitors the activities via telepresence and other displays. Additionally, the functions of controlled mobility, TV and data link communication, latching and energy storage are provided to the robot subsystems by the support subsystems located in each robot and as described in the next section.

A typical functional sequence performed by the Assembler system would include (1) maneuver manipulators to prespecified locations and grip structural elements supplied by a "feeder" device in a known orientation, (2) relocate and orient the elements to proximate locations, (3) using a fine control mode, align the elements to the correct orientations, e.g. with preformed attachment holes aligned, (4) maintain correct orientation and insert fastener element and (5) apply assembly tool to secure the fastener in place. Following completion of assembly segments within the normal Assembler work envelope, other automatic feed devices could be utilized to advance the workpiece to the next position for continuation of the Assembler task on the next segment.

In order to support a selected set of functions required for servicing facility activities, the system could be reprogrammed via the central processor and data bank to load specific maintenance activity sequences for utilization of one or both robot systems in the Technician mode of operation. The Technician functions could include inspection, test, module replacement and replenishment as nominally applied to the satellite servicing area. However spin-off applications may occur in structural and inter-orbit vehicle maintenance. Due to the increased complexity of the tasks, it is assumed that a higher level of knowledge-based operation is required and will be supplied by data downloading from the central processor to the local processor.

One functional scenario for the Technician configuration might consist of an astronaut-aided initial setup in the vicinity of one or more recovered satellites which are attached to a robotic system compatible maintenance platform. The Technician devices would provide the capability for machine vision scanning and tactile sensing of pre-programmed inspection points on the satellite. For example, the test points could be monitored by applying tools or inserting probes. In the case of module replacement, a preset number of fasteners could be disengaged and a new module inserted and secured. For a replenishment task, one operation might involve uncapping fuel connectors, making a fuel line connection and restoring to the operational condition on command after completion of the refueling operation.

## 2.4 Support Subsystem Characteristics

### 2.4.1 Guidance and Control Subsystem

An advanced general-purpose robot for construction,



inspection or maintenance of a space-station must be able to determine continuously its position and orientation in the space-station frame of reference and it must be able to move from one place to another within the space-station structural envelope in response to general "go from A to B, via C" types of command from the central controller. Although the least risky method of moving is to crawl or follow a track maintaining a continuous grasp of the structure by at least one latch, a faster and possibly more economical method, particularly on a very large structure, would be to fly free using some form of reaction jet for attitude control and propulsion. Intermediate between these two methods would be following a cable strung between the two points. The guidance, stabilization and control requirements for a free-flyer or a cable follower are considerably more demanding than for a crawler or track follower, and contingency modes in the event of a partial system failure will require careful study in order to avoid damaging collisions and/or possible loss of the robot - a crawler can simply stop and inform the central controller of its problem, a free-flyer cannot. The final approach to, and latching at the destination of a free-flyer or a cable follower will be a particularly delicate maneuver.

In issuing a travel command to a robot, whether it is a track follower, a crawler, a cable-follower or a free-flyer, the central controller should define a path and/or a time to travel which will avoid collisions with parts of the structure or other robots. However, the knowledge of the central controller may not always be perfect, particularly during construction, and therefore every robot must at all times monitor its own environment. If it differs significantly from what its programming indicates it should be, the robot must stop and/or take avoiding action, as appropriate, and inform the central controller of the discrepancy.

#### 2.4.2 Communication Subsystem

The communications subsystem performs several functions in the postulated robotic application scenarios. These functions include communications links between (1) the robot(s) and the supervisory control station, (2) the various robots working together on a common task (interrobot) and (3) the various interconnected processors comprising the robot control system (intrarobot).

The communications link between robots and supervisory control station provides sensor and status information to the human supervisor to allow real time monitoring and corrective control of robot activity. Because much of the sensor information will be digital video or digital information from other sensor arrays, a wide bandwidth digital communications link will be required. However, the bandwidth requirements for this link can be expected to decrease as the general intelligence and signal processing capabilities of the robot systems increase. A link must also exist in the reverse direction for the human supervisor to send information to the robots. Initially this information would be supervisor override commands. As robot sophistication increases, this communication link would also be used to reprogram robot

functions.

Because many potential robot applications would involve two or more robots functioning together to perform a task, there must be a communications link between them. In addition, since it is anticipated that each robot will contain many processors performing interrelated tasks, there must be communications between the various processors. These interrobot and intrarobot communications link functions are described in more detail in section 3.1.4.3

#### 2.4.3 Energy Transmission and Storage

The energy transmission and storage subsystem provides the prime power to the robot(s) for operation of its various subsystems. These subsystems consists of elements such as computers, motors, electronics, imaging devices, etc. Therefore the prime energy requirement is electrical.

These energy requirements can be satisfied by energy storage devices within the robot, by transmission of energy to the robot or a combination of energy storage and transmission.

This raw electrical energy, either derived from transmitted energy or stored energy, must be converted to a form usable by the robot subsystems. This function is performed by a power conditioning subsystem.

The typical power conditioning subsystem, consisting of components such as inverters, converters, regulators, rectifiers and filters, generates the various voltages and currents required by the other robot subsystems.

A description of alternate energy storage technologies is provided in Section 4.3 of this report.

#### 2.4.4 Latching Subsystem

The space assembly robots will have reaction jets and guidance/control systems, but use of such devices will be limited to unusual or emergency conditions for size, weight, and economic reasons. In general, the robots will utilize their latching subsystem to move from one work area to another and to transfer work loads to the space station. These loads will be seen as minor perturbations to the space station attitude control system.

The latching subsystem must serve a dual purpose. First, it must provide a means by which the robot can be attached to the space station structure. In this way the reactive forces and moments generated by the robot's construction, maintenance, and/or repair tasks will be transmitted to the larger capacity base control system as general disturbance torques requiring a control reaction. Second, the latching subsystem must provide a means of transporting the robot from one work area to another. This requirement is somewhat complex since the structure between two work areas may be large or small, flat or curved, etc. Hence, the latching subsystem may take different forms, depending on the

servicing facility location on the space station. Also, some latching mechanisms (or tracks) may be temporary since they may be needed only during construction. In this case they would be dismantled when construction has been completed. However, other latching mechanisms may be permanent, for example in those cases where they are required for maintenance or repair tasks.

### 3. Robotic System Requirements and Technology Forecasts

#### 3.1 System Requirements and Features

The previous section describes in a general way an application of robotics to the construction and maintenance of part of a space station, emphasizing the rationale for such an activity. In this section the technical issues involved in achieving the stated goals are explored. A general conceptual diagram of a robot system to perform the task is presented and a more detailed examination of a construction task sequence is given. With this as a framework, a set of requirements for each part of the system is developed and the pertinent state of the art reviewed. A brief technology forecast indicates likely future developments under current research directions and helps highlight areas in which additional research is required.

##### 3.1.1 Overall Functional Description

###### 3.1.1.1 System Control and Coordination

Described herein is a robotic system concept which provides a framework for the subsystem analyses and technology forecasts conducted for this study. The global system concept is shown in Figure 3.1.1-1. The lines connecting the functional blocks represent communication paths among processes. The bus-like connection between the central control process and the robotic subsystems is intended to imply communication channels between any pair of attached devices. Throughout, we refer to computer processes rather than to computer or processors to indicate that the actual architecture of the system requires further research. Indeed, some of the communication paths may reside entirely within a single computer system. The major functional units perform the following operations:

1. Task supervision is provided by a human operator at a remote monitor and control station. The supervisor can monitor performance by telepresence, select tasks to be initiated, respond to assistance requests and send manual override commands. As shown in Figure 3.1.1-1, the supervisor's control panel interfaces directly with the central controller which in turn controls the remote units. Further the supervisor console has telecommunication with an Earth based station by which new robot programs may be entered into the system knowledge base. By utilizing programmable task selection and on-call intervention, a wide variety of tasks may be accomplished via data base changes.

The level of monitoring and supervision required for the robotic system operation in various task modes is a direct function of the degree of autonomy assumed for the system and reflects the relative strength of the system knowledge base. A significant objective of this study is to investigate and evolve techniques

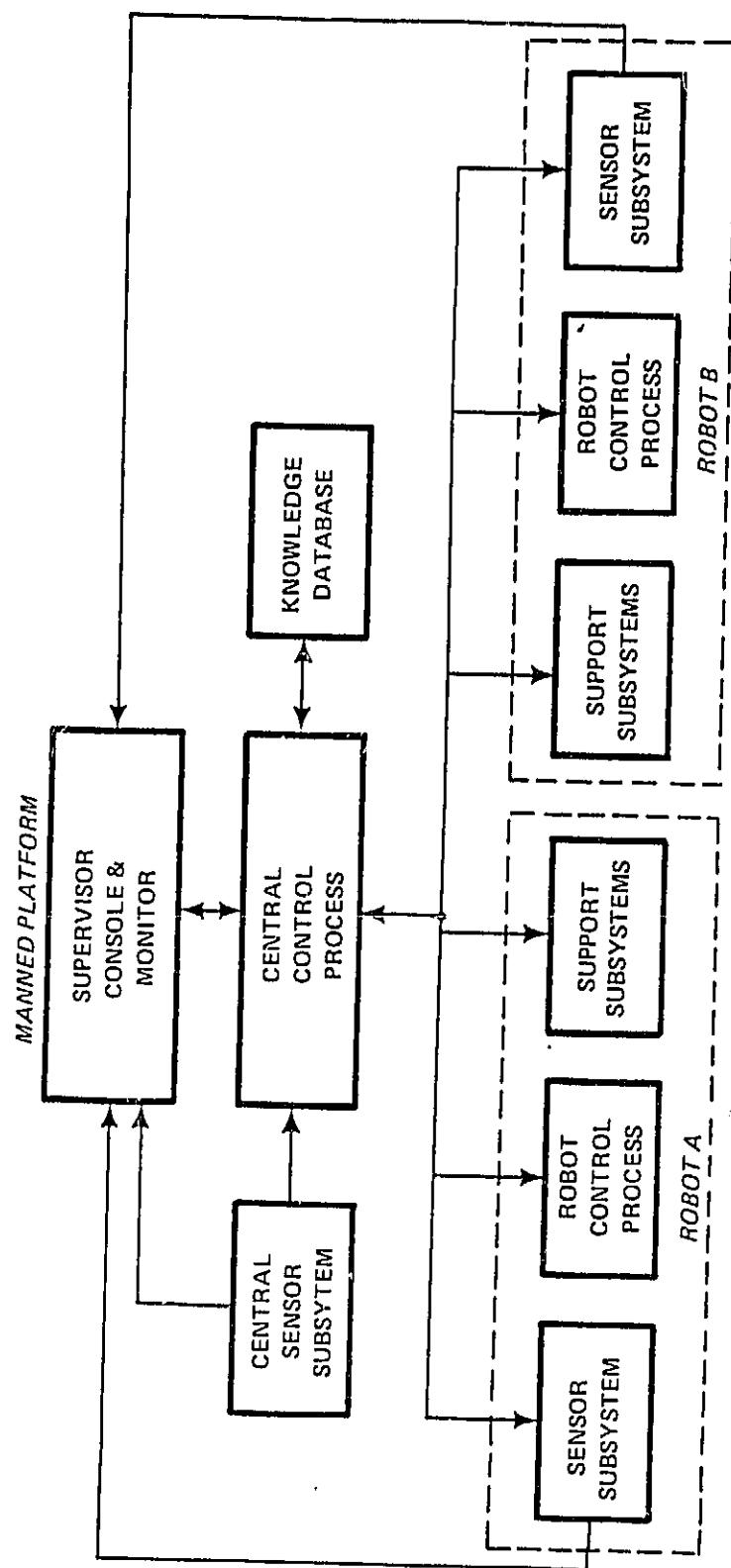


Figure 3.1.1-1 Space Station Robotic System Concept.

which can minimize the human operator task and lead to higher levels of robotic system autonomy within the constraints of practical, realizable systems.

2. The central controller provides the direct supervisory control of the robots such that separate distinct task activities may be carried out by each robot, or cooperating joint activity may be performed. The central controller also monitors and records all station information and provides some level of synchronization among tasks. Flexibility of robot action is provided by stored program call up and communication with the supervisor allows new programs to be entered into the database. Thus the system provides flexibility, expandability and the capability to rapidly substitute and allocate tasks between robots.
3. The local control process for each robot provides the functions of (1) interpreting the central controller commands for the local subsystems, (2) sending status data to the central controller, (3) primary interfacing to the sensors and control subsystems, (4) supporting the computations and signal processing required for the electro-mechanical servo systems (manipulators with end effectors) and sensor data processing, (5) primary interfacing to the support subsystems with command and data links and (6) processing support subsystem signals as-required prior to transfer to the central controller.
4. Within each robot configuration, the sensor subsystems provide for the acquisition of image data and range measurement for use in local robot control as well as a video link to the task supervisor station. Force and tactile sensors are included for use by the local control process. In addition to the primary force or torque generators, the local control subsystems will also utilize position and rate sensors (e.g. shaft encoders and tachometers) as part of the closed loop servo systems.
5. The support subsystems to be implemented in each robot provide for (1) mobility over a small area at low speed in either the "crawler" or self-propelled mode (2) communication with the central controller, task supervisor and other links and (3) electrical energy storage and power distribution. The mobility function is assumed to include subsystems for guidance and control, maneuvering and latching, propulsion, and attitude control.

#### 3.1.1.2 Operational Assumptions

The robotic Assembler configuration assumes dual robots,

each containing its individual robotic and support subsystems for sensing, control, manipulation, communication, power, mobility and other necessary functions. Each robot is constructed with multiple latching arms for attachment to the base and with two or more work manipulator arms for handling tools and workpieces. The robots are normally anchored to the construction platform when handling workpieces during construction and assembly operations but also have a mobility capability to move to a new work site either by crawler or by self propelled mode as directed by the central control process.

With regard to the arrangement of materials used by the robots, it is assumed that automatic feed magazines provide the unassembled work pieces, fasteners and tools within the reach of the manipulator envelopes for utilization by the robots in the assembly task. The structure under assembly is held by a construction fixture which also provides for advance and reorientation of the workpiece as may be advantageous to efficient use of the robotic system. The robots are also assumed capable of translating and rotating to new positions via construction tracks as required during the assembly operations. Other systems would be provided to remove and transport completed assemblies to a storage or usage area and to maintain an adequate supply of construction materials in the magazines.

The general control of the assembly operations by the central process involves directing the coordinated motions of the two robots including mobility commands for translation and rotation of each robot, coarse trajectory commands for each work manipulator of each robot and fine control command sequences for specific manipulators and end effectors as used during the final alignment, fastening and tool application operations. At the robot device level, the commanded sequences are carried out by the local processes utilizing machine vision detection and identification techniques and a variety of contact and noncontact sensors providing feedback data to the inner loop trajectory and fine control servo systems implemented in each robot. At the Assembler system level, the telepresence TV display and the system control panel available to the human supervisor provide an outer loop control capacity to correct for malfunctions and to ensure recovery from complex situations which exceed the normal programmed capability of the dual robot devices.

### 3.1.1.3 Typical Assembler Task Scenario

In preparation for performance of an assembly task, the supervisor would command an initialization and checkout sequence to verify all robotic functions. This would include checkout of the visual and data communication links, automatic checklist monitoring of system parameters such as temperature and voltage levels, verification of downloaded local process programs and issuance of commands to exercise robot manipulators and effectors with automated test sequences for trajectory verification. Additional tests would be performed on the support subsystems to verify the operational readiness of the control, mobility, energy storage and latching subsystems. Following completion of the robotic and

support subsystems checkout and assuming readiness is verified, the central process control would sequence into the work station verification mode where the dual robots verify that all construction material magazines and tools are available, accessible and ready. If magazines require repositioning for robot accessibility, the central process will switch the robots to maneuver mode and request supervisor assistance to grip and reposition the magazines. Once all materials and tools are in place and all functions are verified, the robots are latched in place in the required orientation with respect to each other for the assembly task.

A typical assembly sequence would utilize Robot A to locate, grip and position a new element (for example a truss member) with respect to the workpiece under construction (for example a partially completed beam). Robot A would hold both the workpiece and the new element in a fixed orientation, i.e. the new element is positioned such that matching fastener holes or reference marks in the workpiece and the new element are aligned. Simultaneously, Robot B would position a fastener and fastening tool in the general vicinity of the new element with an initial coarse trajectory mode. When initial alignment is complete, Robot B would switch to fine control mode (for example to insert a pin fastener in the two aligned holes) and would be responsible for the fastening tool application. Once the joint fastening is complete, both robots could utilize vision and tactile sensors to inspect the joint for faults. After inspection and verification, Robot A would locate and grip a new element, Robot B would obtain a new fastener and the process would be repeated until the particular structural assembly was completed.

After completion of the assembly task, the robots would be directed to assist the automatic beam removal system to transfer the beam to a stowage location or to a usage and structure attachment site. After disposal of the assembled beam, the central process would request verification of magazine and tool contents and send commands to reposition the robots for the next task. This will require system reentry to the initialization mode and downloading the appropriate programs to the local processes. In the case where the central process or supervisor commands an operational delay, the robots may be switched to maneuver mode and directed to a storage area pending initiation of the next task sequence.

Although the above task scenario details a nominal set of sequences, the system would also have to include sensing and checking techniques to ensure that contingency situations are also accommodated by the system. For example, additional proximate sensors (scanners or TV cameras) may be required in the work area to provide the capability for robot collision avoidance and for locating parts and tools which may become loose in the zero-g environment. Other types of complex faults requiring supervisor intervention could include missing tools or components, broken tools, improper joints or inspection failures of any type, failure of latching or mobility subsystem requiring robot rescue by astronaut EVA and other general loss of robot control by power or



device failures.

### 3.1.2 Robot Structure and Control

The space station tasks described in Section 3.1.1 above require dexterity similar to a human arm and hand. Extensive fixturing, as is done in many earth based robot manufacturing tasks will be infeasible in space. Rather, a second (and perhaps third) robot will be used as a general purpose fixture. This requires the execution of several processes (some independently and some simultaneously) for task completion. Some of these processes are real-time in nature such as servoing the joint motors of the space robots in gross motion and fine motion control, coordinating the motion trajectory among several robots, and interacting with the supervisor to complete the tasks in a coordinated supervisory control mode. Others are relevant-time because they are time-consuming but not time-critical and require processing a large amount of information, such as trajectory planning and visual recognition and inspection of mechanical objects. Speed, however, is not critical. Rather, key requirements are reliability, accuracy, autonomy, robustness of activity, and the ability to recover from exceptional conditions.

The implication of these requirements on the space robot structure and control may be divided into three areas: the manipulator arms, the hands and end effectors, and the control systems.

#### 3.1.2.1 Manipulator Arms

An industrial robot is a general purpose manipulator consisting of several rigid links connected in series by revolute or prismatic joints. One end of the chain is attached to a supporting base while the other end is free and equipped with a tool to manipulate objects or perform assembly tasks. The motion of the joints results in relative motion of the links. Mechanically a robot is composed of an arm (or main frame) and a wrist subassembly plus a tool. It is designed to reach a workpiece located within its work volume. The arm subassembly normally provides three degrees of freedom movement so that the combination of the movements will place or position the wrist unit at the workpiece. The wrist subassembly unit usually consists of three rotary motions. The combination of these motions will orient the tool according to the object configuration to facilitate pickup. These last three angular motions are often labeled as the pitch, yaw and roll. Hence for a six-joint robot, the arm subassembly is the positioning mechanism, while the wrist subassembly is the orientation mechanism.

Many commercially available industrial robots are widely used in manufacturing and assembly tasks (such as simple material-handling, spot-arc welding, parts assembly, paint spraying, loading and unloading numerically-controlled machines), in space and undersea exploration, in prosthetic arm research, and in handling hazardous (e.g. radioactive) material. These robots, which exhibit their characteristics in motion and geometry, fall

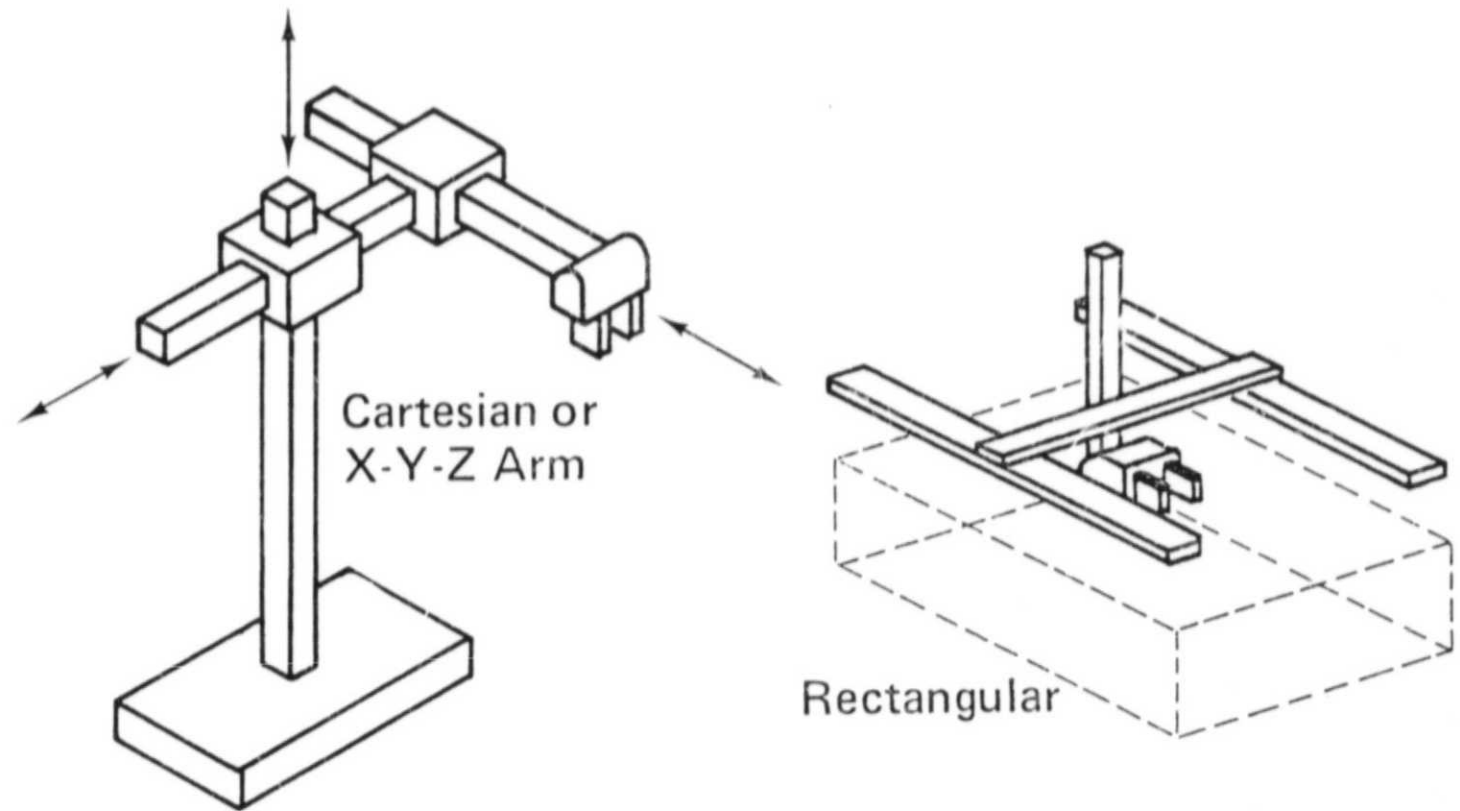
into one of the four basic motion-defining categories (see Figures 3.1.2-1 through 3.1.2-4):

- (1) Cartesian coordinates (three linear axes), (e.g. IBM's RS-1 robot and Sigma robot from Olivetti)
- (2) Cylindrical coordinates (two linear and one rotary axes), (e.g. Verstran 600 robot from Prab)
- (3) Spherical coordinates (one linear and two rotary axes), (e.g. Unimate 2000B from Unimation Inc.)
- (4) Revolute or articulated coordinates (three rotary axes). (e.g. T3 from Cincinnati Milacron and PUMAs from Unimation)

Due to the environmental constraints in space applications, it may be useful to design redundant robots which have more than six degrees-of-freedom. The use of such robots in space applications requires further study of kinematic analysis and control strategies. Furthermore, from the application point of view, one would like to design a lightweight robot with high payload capability. This requires investigations into the use of composite materials to strengthen the rigidity of robot arms and improve the flexibility effect of the links. For industrial applications, robot arms are moved at their maximum speeds to increase production. However, for space robots, it may be more important to control the applied joint torques than speed. In particular, the space assembly robots may require monitoring and generating the required torques for the given assembly tasks. In order to complete the required space station assembly tasks, space robots will need sensors to measure vital feedback sensory information such as link positions and velocities, interaction forces between the assembly interfaces, tactile sensing of the object gripping force and visual inspection of the assembled parts. This sensory information will be vital for generating joint torques to servo the robot arm.

#### 3.1.2.2 Gripper and End-Effectors

The motion of the manipulator joints positions the end-effector or the gripper at the work-piece for performing desired tasks. The most commonly used gripper in the industry is a pneumatically-controlled parallel jaw. Quite often, the desired tasks require more sophisticated end-effectors or tools, such as a multi-joint multi-fingered hand. Salisbury [Sa180] designed a three-joint, three fingered hand for picking up irregular objects and performed a kinematic analysis on this prototype hand. A group of researchers at the University of Rhode Island have looked into the design of end-effectors and have designed different grippers for various tasks. Besides designing unique task-specific grippers, it is also important to look at the mechanisms for tool changing. Since space robots may perform different tasks requiring tool changing, a simple tool changing mechanism must be devised to (1) ease the process and (2) ensure that calibration of the hand is still valid. Furthermore, the gripper may also be designed to



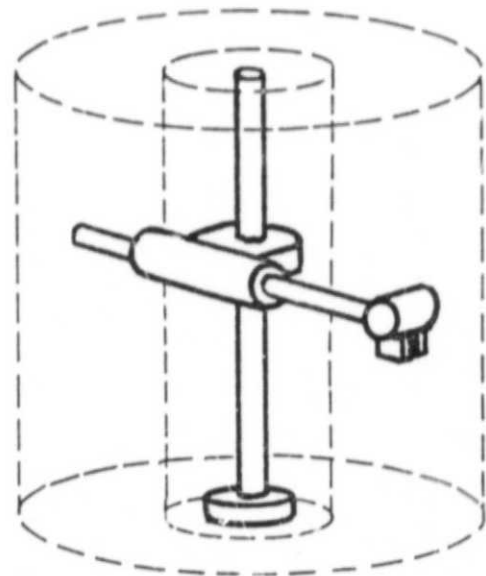
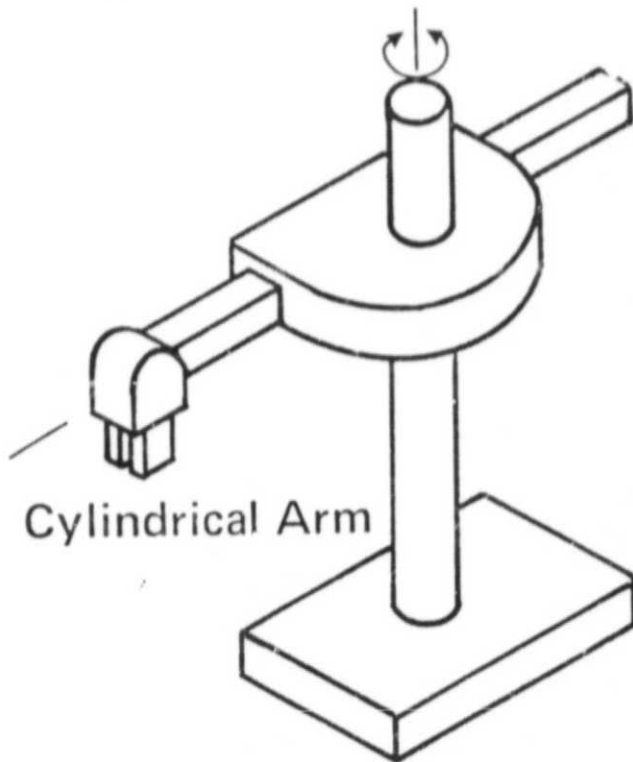
Advantages:

1. Greatest resolution.
2. Control of the joint motors is very simple.
3. Good obstacle avoidance.

Disadvantages:

1. Large amount of structure.
2. Restriction on compatibility with other arms in a common workspace.
3. Three linear axes make the mechanical design more complex than other robots.
4. Needs more floor space.

Figure 3.1.2-1 Cartesian Coordinate Robot



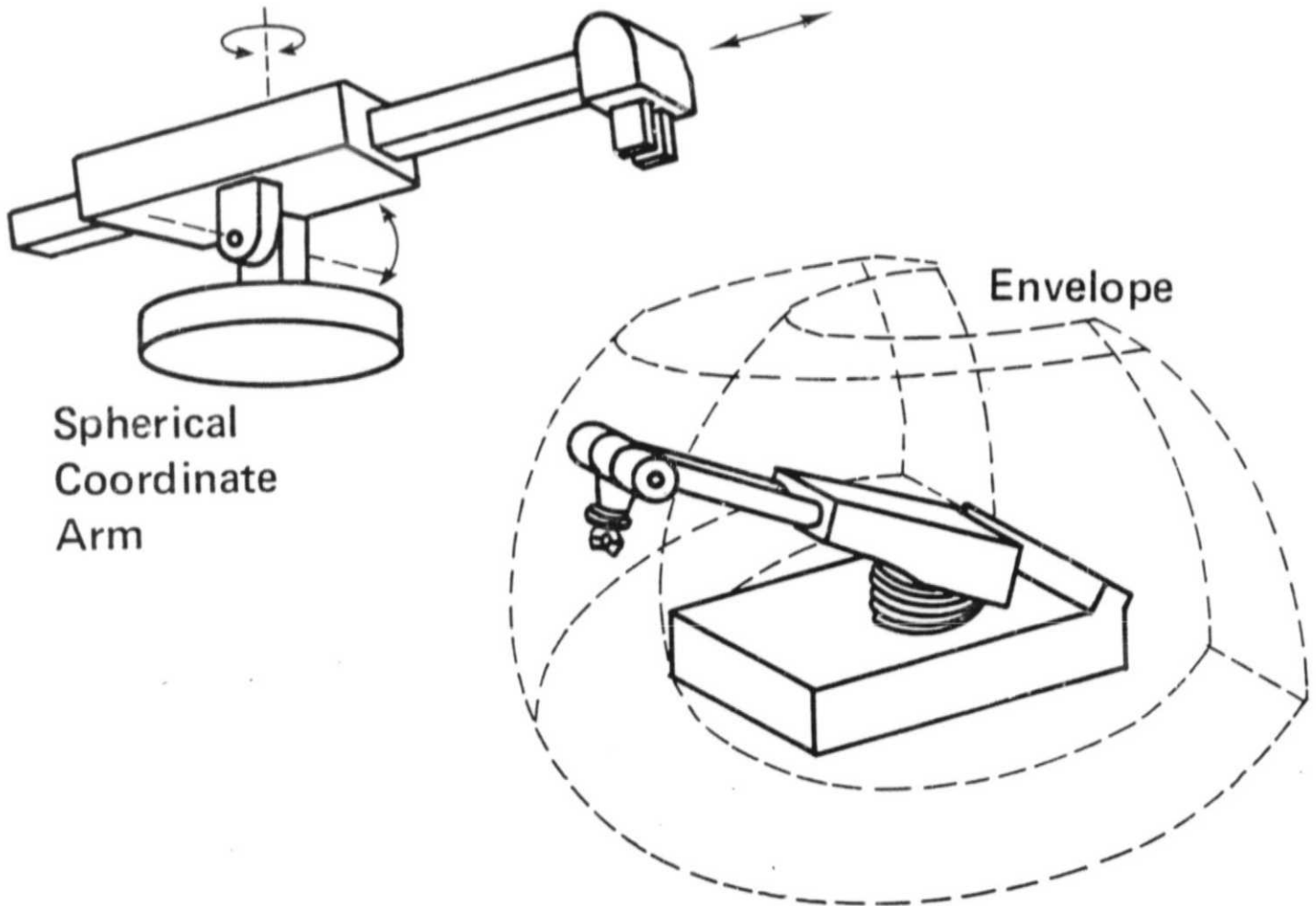
Advantages:

1. Control of the joint motors is very simple.
2. Good obstacle avoidance.

Disadvantages:

1. Large amount of structure.
2. Restriction on compatibility with other arms in a common workspace.

Figure 3.1.2-2 Cylindrical Coordinate Robot



**Spherical  
Coordinate  
Arm**

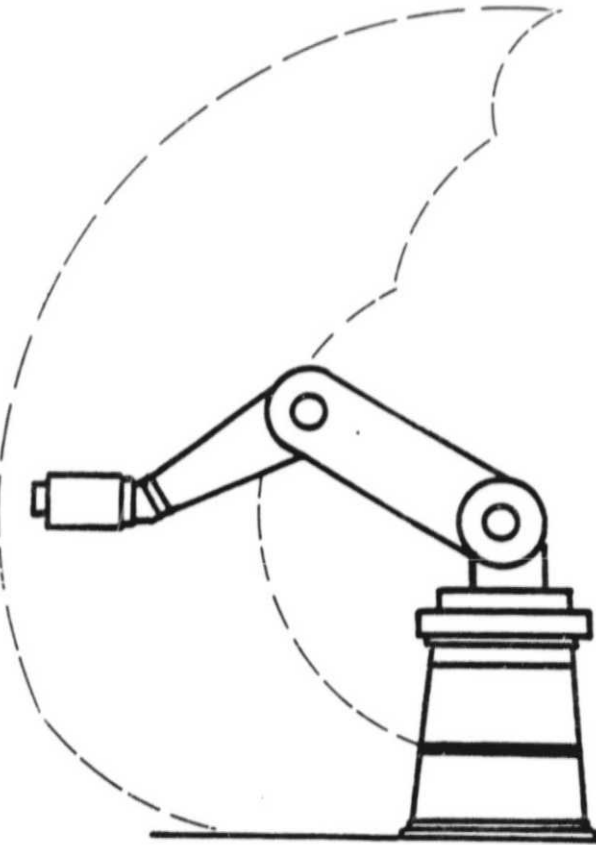
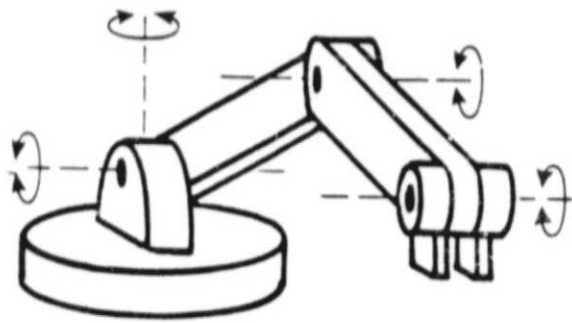
**Advantages:**

1. Lowest weight and minimum structure.
2. Good resolution because the joints give mutually perpendicular contributions to position error.
3. Short joint travel for many applications.
4. Compatible with other robots working in the common workspace.

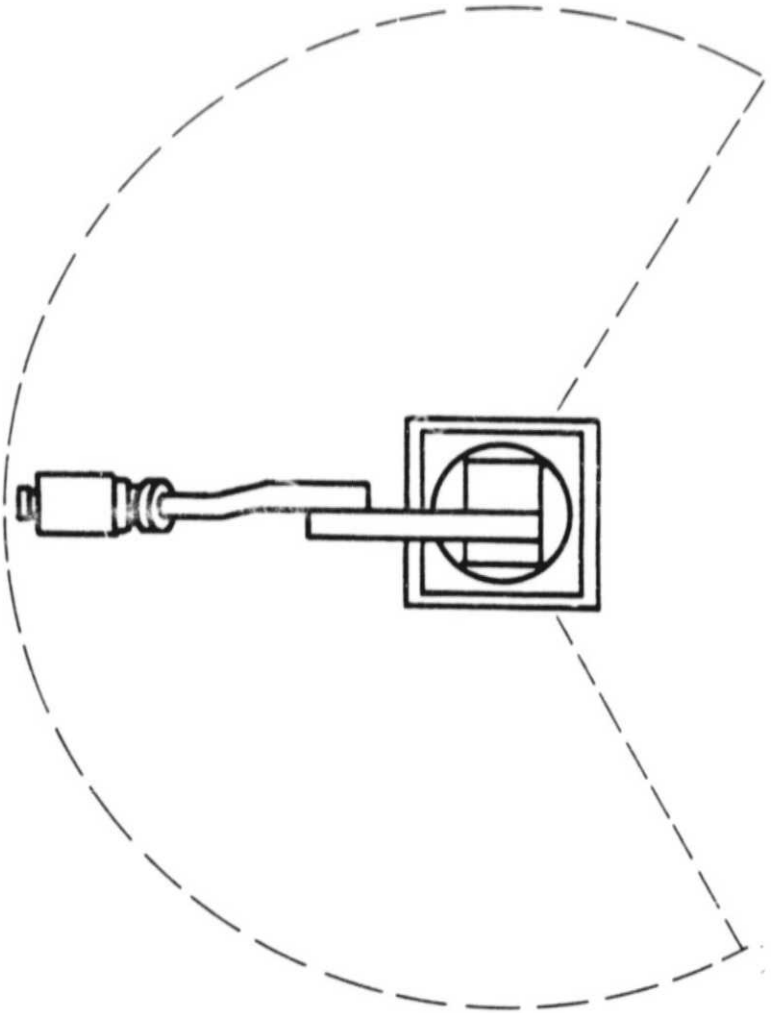
**Disadvantages:**

1. Large and variable torques on joint 2 and 3 of the arm and counterbalance problems.
2. Ability to avoid obstacles is limited.
3. Position error is proportional to the radius at which the arm is operating.

**Figure 3.1.2-3 Spherical Coordinate Robot**



Side View



Top View

**Advantages:**

1. Has the most flexibility to reach over or under an object.
2. Compatible with other robots working in the same common workspace.

**Disadvantages:**

1. Exhibits poorest accuracy given the same workspace and maximum sensor resolution. The position precision is proportional to the operating radius.
2. Large and variable torques on joint 2 and 3 of the arm, counterbalance problems and limited obstacle avoidance.
3. The large outboard joint (joint 3) increases the arm's moment of inertia.

**Figure 3.1.2-4 Revolute Coordinate Robot**

incorporate some type of non-contact sensors (visual or proximity), for example to provide path guidance and product inspection capability.

#### 3.1.2.3 Control Systems

Most industrial robots find their applications in material handling and spot welding and as such do not require sophisticated control systems. The user usually controls such motions by moving the robot arm through space and recording a set of positions along the path. The robot arm is then guided to duplicate or playback the reference motion. Such duplicate motion control gives a fairly accurate position control because the repeatability of most robot arms is quite good. However for space robots, most of the assembly tasks may not be compatible with the duplication technique because of the difficulty in teaching the robot arm with flexible links. Hence a high accuracy position control requirement augments the impetus for designing a high performance controller for space robots. Due to the link flexibility and the expected tolerance of the assembly tasks in space station, sensory feedback information will be needed to compute the correction torques at every instant of sampling. As discussed in the sensor section, internal sensors such as potentiometers, tachometers, and encoders are used to measure the position and velocity of the robot arm while force, tactile and visual sensors are used to sense the environmental conditions for completing the task.

It is worth noting that in an assembly task cycle, the robot arm may have to move several parts of different inertias and may require adaptive control on the part of the control system to compensate/modify acceleration. Various adaptive controls are found suitable for such motion control, in particular the resolved motion adaptive control [LeL83].

#### 3.1.3 Sensing and Machine Vision

The robots to be launched in space will require extensive sensing to provide feedback to the automatic robot control systems and to make position changes relative to what it senses. We can thus examine the robot sensor requirements from four points of view: assembly, inspection, system monitoring and data processing requirements. The latter is pertinent because sensor data processing will be one of the prime factors influencing the computer requirements.

Besides the physical sensors themselves the mission needs will be developed for processing and analyzing the sensor data. We shall discuss below the mission requirements for sensors and the algorithms used for processing the sensor data.

##### Assembly Tasks

At least three basic types of sensors are expected to be needed for the assembly tasks (aside from the usual position and rate servo feedback) as represented by the following categories:

- o vision sensors
- o force sensors
- o tactile sensors

Vision sensors will assist in the recognition of the part(s) from the magazine, in gripping, and in gross motion control. It is well agreed that force sensing is needed to accommodate small changes in planned work positions, and accomplish assembly operations. In a recent study of 41 assembly tasks by Harmon it was concluded that tactile sensing was needed in 83% of the cases. It is likely then that tactile sensing will be required in space applications as well.

Other sensors which may be required are global position sensors for the mobile robots and proximity sensors to detect lost parts or foreign objects in the vicinity.

### Inspection Tasks

The beambuilder robot must be capable of inspecting the joints after assembly. This type of inspection will include inspection for alignment of joints and making sure all fasteners were installed. Some strength testing is expected. To achieve the latter perhaps an X-ray inspection may be used.

The inspection algorithms will include part recognition and orientation. The system will use the ranging information to feed back to the robots where the parts, magazines, and beams are located. The ranging information should be continuously available. The inspection algorithms will have to be able to cope with shadows and other visual artifacts. The inspection system will also act as a guard in case parts break loose and drift into the work area. This could be a separate visual sensor that observes a larger view of the work area and sends "halt" signals to the task supervisor if a drifting part is detected.

### System Monitoring

In addition to a need for a variety of sensors for the automatic operation of robot controls, sensor information must be available both to resident human supervisors and possibly via telemetry to ground based human monitors. This in turn will require the deployment of visual sensors in positions from which the total work area and robot area may be viewed. They will undoubtedly need to be controllable with respect to viewing direction, and probably require controllable zoom lens.

All of the data obtained from the active robot sensor should be made available to the supervisor. In many cases this will require processing and graphic display in order that the data may be intelligently interpreted.

### Data Processing Requirements

The machine vision computer system must be capable of handling large amounts of image-like data in relevant time. The



visual sensors will produce one 512 x 512 frame of data about every tenth of a second while a tactile sensor can be expected to produce 8 x 8 or 16 x 16 images at approximately a 100 Hz rate to provide the required accuracy. The system must also be capable of high speed floating point operations and will need several megabytes of image storage just for the acquired data. The system should be capable of doing window operations at video rates and using the ranging information to solve the depth problem.

#### 3.1.3.1 Vision Sensor

Vision sensors, in particular, may appear in several forms. The robotic support systems will need to provide (or at least be able to control) the source of "illumination" for these visual sensors. The vision sensors will include visible light as well as nonvisible light sensors so that a human operator can stay in the loop when necessary.

The visible light sensors should be capable of sensing different colors. A color CCD camera will probably be the desired system for this requirement. The system should be capable of digitizing a 512 x 512 image with 8 bits of gray level resolution for each color. This requirement could be reduced initially by using a gray scale (black and white) camera with automatic zoom lenses. The lenses should also have automatic iris capability. In some assembly or inspection tasks it may be necessary to have more than one view of the work area. This will require more than one camera or set of sensors. While it will probably not be necessary to process the sensor data in real-time it may be necessary to have a good deal of memory to store all the images. For the assembly operations, depth or range information must be available and a range sensor with millimeter accuracy may be needed. Range sensing can be accomplished by using radar, laser light or structured lighting. The sensors must be robust enough so that they can handle some of the environmental constraints of space flight; e.g. the CCD sensor must be capable of tolerating very bright light from small sun angles for long periods of time. The sensor system must be capable of handling extraneous information from variable shadows and sunlight effects and the illumination sources must have the capability of being reconfigured for each of the tasks.

#### 3.1.3.2 Force Sensors

Most assembly tasks require operations such as the insertion of close parts, following an edge (for arc welding), opening hinged covers, and tightening bolts and screws. In these operations, sensing the gripping forces on the workpiece and the forces that develop between the end-effector of the manipulator and its assembly interface are very important and provide vital feedback information to the servo control for guiding and controlling the robot in completing its task.

Such "contact" forces can be extracted for control purposes by compact, reliable sensors. Sensing the forces developed at the assembly interface can be realized by (a) a six-axis wrist force sensor which is tailored to be mounted at the

last joint's mounting plate of the robot, (b) a pedestal sensor, or (c) monitoring the armature current of the joint motors. Furthermore, for close-parts insertions with chamfer edges, a passive device developed at the Draper Laboratories called remote center compliance can be used.

Most of the above force sensors are made of mechanical components. Hysteresis and coupling between the sensing components may be present and require further processing or decoupling of the sensory information for control purposes. Using today's technology as a base, compact, reliable, sensitive, solid-state force sensors with transducers and processing components in a single chip may be developed for space robots to improve their performance in assembly task operations.

### 3.1.3.3 Tactile Sensor

For the structural assembly task, sensing the gripping forces inducing slippage between the gripped object and the inner surfaces of the gripper provides vital information for properly acquiring and transferring the object to its designated location. Such sensory information can be extracted by tactile sensors which are usually mounted on the inner surfaces of the gripper. Unlike the wrist force sensor, the tactile sensor senses the pressure present between the gripping surface areas. These forces are usually much smaller than those sensed by the wrist force sensors. VLSIC technology can be utilized to develop high resolution tactile sensor arrays (e.g. 64 x 64 or 128 x 128 sensing elements). Current resistive type commercial tactile sensors have low resolution (8 x 8 element arrays) however, capacitive high resolution tactile sensors may be more desirable for space robots. Similar to the visual image case, processing the tactile images in real-time will require large memory space and fast CPU hardware. Unlike parts recognition, the tactile images have to be processed quite rapidly to ensure proper handling and transferring of the object in the assembly or maintenance operation.

### 3.1.4 Computers, Language and Communications

#### 3.1.4.1 Requirements for Space Computers

The requirements for the computers implementing the central and local processes which will control the robot and monitor systems are derived from both the computational performance needed and the harsh environment in which the system must operate. They fall in the following categories:

- o high reliability
- o low power consumption
- o high computational capabilities
- o large memory
- o real time input/output (I/O) structure
- o distributed system operational compatibility.

Since space robot computers may be expected to operate frequently for long periods without receiving service, they will

have to be reliable and require low power consumption. Even with fast advancing VLSI technology, it seems difficult to manufacture computer components that completely satisfy expected stringent reliability requirements. Note that NASA constructed two ultra reliable computers for civilian aircraft, namely FTMP [HOP78] and SIFT [WEN78] which are intended to meet a specified failure rate of  $10^{-9}$  per ten hour mission. This rate was specifically for civilian aircraft in the 1990's which typically have much shorter mission lifetimes than space applications.

Suggested approaches to the reliability issue regarding a specification similar to the above include utilizing expected improvements in component reliability and design improvements through use of multiple hardware and software modules providing redundancy, reconfigurability, fast fault detection, non-interference repair, and the like.

It is important to observe that reliability alone can not meet all the system needs. Particularly, space computers must have high throughput capability to cope with the need of real-time computations. Computers may be regarded as malfunctioning simply because they do not respond fast enough to environmental triggers [KrS83]. Note that this may cause a system catastrophe just as hardware/software component failures do.

In order to obtain high throughput, one may have to use multiple processors and also partition computation tasks into loosely interacting subtasks to be executed on separate processors.

Multiprocessors have been receiving wide and extensive attention from the computer architecture community as a viable solution to both the reliability and the throughput mentioned above. This popularity is mainly based on their potential for (a) providing improved reliability thru the inherent redundancy with component multiplicity, and (b) computation speedup (hopefully close to linear speedup) with cooperation from multiple processors. However, the efficiency of multiprocessors in providing improvements in both reliability and throughput heavily depends on their interconnection/intercommunications, synchronization, I/O interface, operating systems, task partitioning, etc. Nevertheless, multiprocessors are a strong candidate for space robot applications. Consequently, the foregoing issues must be carefully examined and resolved so that the performance improvements due to multiprocessor utilization are well defined and verified.

Another important space computer aspect is the structure of space computation tasks. The more we understand the structure, the more effective a computer architecture can be defined. This is apparent in view of the fact that a computer may not function equally well for different computations. Thus, knowledge of the computation structure can be transformed into a computer structure with better reliability and higher throughput. Structuring the space application tasks should be done in conjunction with expertise in various design areas such as computer architecture, control theory, numerical analysis, software programming and

related disciplines.

Finally, any space computer system should have the capability of providing easy interface to users (both astronauts and regular programmers) as well as to other system components (e.g. other real-time sensors and actuators, computers, displays, intelligent peripherals, etc). The former requirement is strongly influenced by the robot language considerations discussed below.

#### 3.1.4.2 Robot Languages

The development of a powerful, reliable, compact, flexible computer system for the Space Robotics application is not sufficient, however. It must be possible to program the computer and communicate with it readily if it is to be effective. The attachment of devices with real time control tasks to the computer adds considerable complexity in the programming area.

One of the most important advantages offered by robots is the flexibility to use a robot for many different purposes. Consequently, it is essential that robots have the capability of switching jobs without requiring excessive time or alteration of basic structures of the system at hand. The capability can only be obtained through efficient programming means provided by robot languages (RLs). Conventionally, RLs have been developed on an ad hoc basis as a need for them arose and are, therefore, geared toward specific robots and their applications. If continued, this trend could result in unlimited proliferation of RLs and thus present many potential problems, both economic and operational (e.g. the space system user training complexity). To alleviate this problem, this report investigates global issues and requirements and formulates recommendations for future effort.

There are a number of issues to be resolved before any future RL is designed. These issues include:

- 1) user classification
- 2) robot language evaluation
- 3) exception and error handling requirements
- 4) interactions with robots and their environment
- 5) versatility for various robots and computers.

The first issue is based on the fact that in general the requirements for an RL depend on the type of users. For example, an RL for the end users (i.e. astronauts) must emphasize naturalness for applications, whereas for the system developers in space robotics it should contain capability of programming low-level control algorithms, hardware/software module interfaces, etc. This need naturally leads to the decomposition of the RL into a multi-level structure which is suitable for different classes of users. The RL has to provide various detail levels corresponding to each class of users.

The second issue is needed to provide an objective means of evaluating and then improving design of RLs. Since features of the RLs are qualitative in nature, evaluation methods tend to be subjective. This is inherently similar to the problem of generating software metrics to measure efficiency and effectiveness of general purpose programming languages. However, one may be able to devise a better method for evaluating RLs due to their special purpose use.

The third issue is concerned with the very nature of supervised semi-autonomous robot systems. This aspect is particularly important for space applications since the robot systems will have to operate for most of the time without human attendance. Robot systems must be able to handle some abnormal events with their own intelligence. The issue may be divided into two subissues: one is exception handling which is analogous to that in general computer programming. That is, abnormal events internal to the systems (e.g. software exceptions) belong to this category. The other subissue is abnormalities external to the systems. For example, dropping an object accidentally, unexpected changes in the robots' environment, picking up a wrong object, etc. are certainly the cases that this subissue has to deal with. Any solution to this has a close relationship with the next issue, i.e., interaction with environment. The fourth issue is related to external communications. We expect that any space application, particularly space station construction will require multiple robots, various tools and parts, fixtures, transport machinery, etc. Changes in these components must be monitored and the consequent actions must be taken in real-time. In other words, real-time data acquisition, processing, and actuation are mandatory. Of course, these actions are to be formulated and then performed synergistically among all processes in the system. Hence, flexible but fast communications capability among these heterogeneous components have to be included in robot languages.

The final issue considered here is the transportability of the robot language. Any space RL should be usable for various robots, computers and other intelligent machinery. This may be much harder than the transportability problem for general programming languages since RLs have to deal with system components that are vastly different from one another. One way to look at this issue is to delegate intelligence to local processes as much as possible. If each system component is highly intelligent, then it can be made to understand one or more universal robot languages thus meeting the requirement of transportability.

#### 3.1.4.3 Communication Requirements

An indispensable part of the overall space robotic system configuration will be data communications between various pairs of processes. Many of these are obvious. The robot control process must communicate with its sensor processes, the central control process must communicate with each robot process, etc. Others, however, are not so obvious. When working on a coordinated task, Robot A may need to synchronize with Robot B, or access sensor

information from Robot B. For this initial study, therefore it is assumed that there must be digital data communication between every pair of processors. In other words, there must really be a digital communication network interconnecting the processors in the system. In addition, there must be analog vision transmission between the monitor camera and the supervisor console.

The digital data transfer must be extremely reliable as the goal is to eventually operate in an unattended mode for long periods. Thus the entire operation will depend upon accurate data. The communication system must therefore include a reliable error checking protocol. The data rate required is less certain. If complete images are transferred, then on the order of  $2 \times 10^6$  bits are needed per image and the system should be capable of  $10^8$  bits per second. However, if a more complete system design reveals a need to transmit only coded images or features extracted from images, the required data rate might be significantly less than this figure. Another factor will be the rate at which the entire system operates. As noted earlier, the issues of reliability, accuracy and energy consumption are much more important than speed. Consequently data rates in the range of  $10^7$ - $10^8$  bits per second seem adequate.

The implementation of a computer communication network in a space application will itself introduce many system requirements. However, as those are dependent upon the particular implementation chosen, further discussion is deferred to section 3.2.3.3 in which alternative strategies are discussed.

### 3.1.5 Expert & Knowledge Systems

Artificial intelligence (AI) and knowledge based systems will be needed and utilized by the proposed Space Robotics systems. Requirements in this area, however, are not as readily quantified as in other areas since they are more a matter of degree resulting from tradeoffs. The goal is to develop a semi-autonomous system with the level of autonomy actually achieved determined by the sophistication of the knowledge based techniques available.

Two generations of systems are considered, the assembly system described above, and an automatic technician maintenance service, the latter requiring appreciably more sophistication than the first. The requirements in the knowledge based systems include the development of the following AI capabilities:

- o Problem solving
- o Knowledge based techniques

which are discussed in the sections that follow.

#### 3.1.5.1 Problem Solving

##### 3.1.5.1.1 Distributed Problem Solving

This report assumes that two or more robots will cooperate

in an assembly task in a Space Station construction area. Each robot may have its own sensor system for determining the orientation and location of parts. There may be some sensors directly under control of the control computer. Though each robot has its own processor, the overall coordination is achieved by communication with the central process control computer. The control computer must have a global picture of the system at a given instant to decide the next step. Since each robot has a local processor and can act independent of other robots, the overall coordination will require communication of the location of a robot and of the orientation and location of a part to the control computer. The control computer has a database that represents the global state of the system and a knowledge base of the environment, tasks to be performed, all parts and tools, and exception condition management. Each robot has a database representing only its state. The task of selecting the next step in this environment faces all problems encountered by distributed artificial intelligence research.

In multi-robot systems, each robot will perform a sub-task as determined by the local processor via an instruction from the central control computer, and then communicate the consequence of the subtask to the control computer. The location and orientation of a part, important features of the part, etc. will be sent to the control computer. Since the control computer has a global picture of the state of the system, it is in position to interrupt action of local processors or determine the next sub-task to be performed.

#### 3.1.5.1.2 Spatial Reasoning and Motion Planning

In an operational space station assembly or servicing task, the number and location of objects will be changing from time to time and the robots will have to function in diverse situations. This requirement can only be satisfied if the robotic systems have the capability to adapt their operation sequence to the environment.

Two most crucial capabilities required for the space robots to function in a semi-controlled, environment are spatial reasoning and motion planning. At a particular time, location of objects may be known. A problem to be solved by a robot may be how to move an object from a location  $x$  to another location  $y$ . To solve this problem the robot system has to find (1) whether on a particular path there is enough space to move the object and (2) which path is best. The first problem requires that the controller should be able to reason using physical dimensions about the plausibility of a path. The second problem requires finding paths for the movement in a given configuration and if there are no paths then deciding what actions (e.g. like moving some objects) can be taken to make paths.

#### 3.1.5.2 Knowledge Based Systems

##### 3.1.5.2.1 Assembly Activities

One way of representing a structural assembly task using

robots and sensor systems is through procedural knowledge representation. Primitive robot operations will be identified; some basic operations using robot motions and their relations to sensors will be represented using the primitive operations. An assembly task will be simulated to represent the concept of achieving a set of goals as in an artificial intelligence language like PLANNER or RAPT. This representation may be considered as a program for an assembly task. The program will, however, be different from a program written using today's robot programming languages. By imbedding control knowledge in the program, the system will be given the ability to handle a new situation gracefully.

The robot task operations specified using procedural knowledge will be satisfactory for simple assembly tasks and these systems will allow desired interaction with the sensors. Programs for different assembly tasks can also be stored at the control process. To initiate a new assembly task, the program can be sent to the supervisor from the on-board data base.

#### 3.1.5.2.2 Inspection and Technician Activities

The task of inspection is significantly more complex than the task of simple assembly. A knowledge base for numerous possible faults, both structural and physical, must be stored for each system. For a simple component this task may not require a sophisticated knowledge base; a model may be enough. As the component grows in complexity, the complexity of the inspection grows exponentially. Sensor systems will be required to give very precise information about the location, orientation and similarity of the parts with their model.

In some cases laser ranging, IR or X-ray sensors may be required to detect internal faults. It may be possible to develop techniques, based on X-ray sensors, to find the presence or absence of an internal part in a complex assembly. Tactile sensors may be used to inspect smoothness and other properties of a surface finish using much less computational effort as compared to vision sensors for the same task.

The inspection task for completed beam assemblies will be less complex. Control of the camera location to obtain images from different viewpoints may be required, however. Using a model of the beam, important viewpoints will be determined to inspect the assembly from all sides.

The technician will employ sophisticated expert systems, distributed problem solving, and may interact with the ground station. However, it is anticipated that a significant interaction level will be required between the supervisor and the technician system.

One servicing facility task will be to detect faults. The technician will have to identify causes of the fault by analyzing the knowledge base, determining methods to fix the faults, and finally using stored knowledge to complete the repair.



The technician may have to interact with the ground station in case of a fault with low confidence diagnosis. The ground station may either give some hints or may take over the diagnosis task. Similar interaction may also be needed for repair activities.

In many cases, the faults may be structural. Faults in assembled components may not be detectable by inspecting them from the outside and disassembly may be required for fault detection leading to a significantly increased task complexity. The assembled component may be disassembled systematically, and each part inspected and properly placed for later reassembly. As can be seen, this kind of inspection will require a program to disassemble using multi-robots, models of good components, and knowledge to check each part using vision or other sensor based methods.

All fault detection and repair activities will be performed with some degree of reliance on expert systems. Knowledge about possible sources of faults, based on tests, will be stored with methods to repair faults. The expert system will instruct robots and sensor systems to perform pertinent tests. Based on the test results reported by the sensor systems, the next action can be decided and performed. In those cases unfamiliar to the expert system, the ground station will be contacted. The new situations will help the expert system in acquisition of knowledge which can be applied if the same situation recurs.

### 3.2 Current Capability Analyses and Research Directions

Herein we will review the current state of the art defining robot systems in each of the areas necessary for the robotic space station construction system and describe the current research direction in these areas. We also consider the relevancy of both current technology and the known research directions to the issues associated with the application of space robotics to the construction and servicing facility tasks.

#### 3.2.1 Robot Structures and Control

The current technology descriptive work reviewed in robot structure and control is divided into four areas:

- o kinematics
- o dynamics
- o motion control
- o robot structures

each of which is discussed and described in the sections which follow.

##### 3.2.1.1 Kinematics Analysis

The kinematics problem deals with the analytical study of the spatial configuration of the robot arm as a function of time without regard to the forces/moments that cause the motion. In particular, it is concerned with the relations between the joint-variable space and the position and orientation of the

end-effector of a robot arm.

The kinematics problem usually consists of two subproblems, i.e. the direct and inverse kinematics. The direct kinematics problem is to find the position and or orientation of the end effector of a manipulator with respect to a reference coordinate system, given the joint variable vector

$$q(t) = [q_1(t), q_2(t), \dots, q_n(t)]^T$$

of the robot arm and the various geometric link parameters, where  $n$  is the number of degrees-of-freedom. The inverse kinematics problem (or arm solution) is to calculate the joint variable vector  $q(t)$  for positioning the end-effector of the robot arm at the desired position with the desired orientation, given the position and orientation of the end effector with respect to the reference coordinate system and the various geometric link parameters. Since the robot servo system requires the reference inputs to be in joint coordinates and a task is generally stated in terms of the Cartesian coordinate system, controlling the position and orientation of the end-effector of a robot arm to reach its object requires the understanding of the kinematic relation between these two coordinate systems.

To describe the translational and rotational relationship between adjacent links, a Denavit-Hartenberg matrix representation [DeH65] for each link is used and shown in Figure 3.2.1-1. From Figure 3.2.1-1, an orthonormal coordinate frame system  $(x_i, y_i, z_i)$  is assigned to link  $i$ , where the  $z_i$  axis passes through the axis of motion of joint  $i+1$ , and the  $x_i$  axis is normal to the  $z_{i-1}$  axis pointing away from it, while the  $y_i$  axis completes the right hand rule. With this orthonormal coordinate frame, link  $i$  is characterized by two parameters:  $a_i$ , the common normal distance between the  $z_{i-1}$  and  $z_i$  axes, and  $\alpha_i$ , the twist angle measured between the  $z_{i-1}$  and  $z_i$  axes in a plane perpendicular to  $a_i$ . Joint  $i$  which connects link  $i-1$  to link  $i$  is characterized by a distance parameter  $d_i$  measured between the  $x_{i-1}$  and  $x_i$  axes and a revolute joint variable  $\theta_i$  which is the joint angle between the normals and measured in a plane normal to the joint

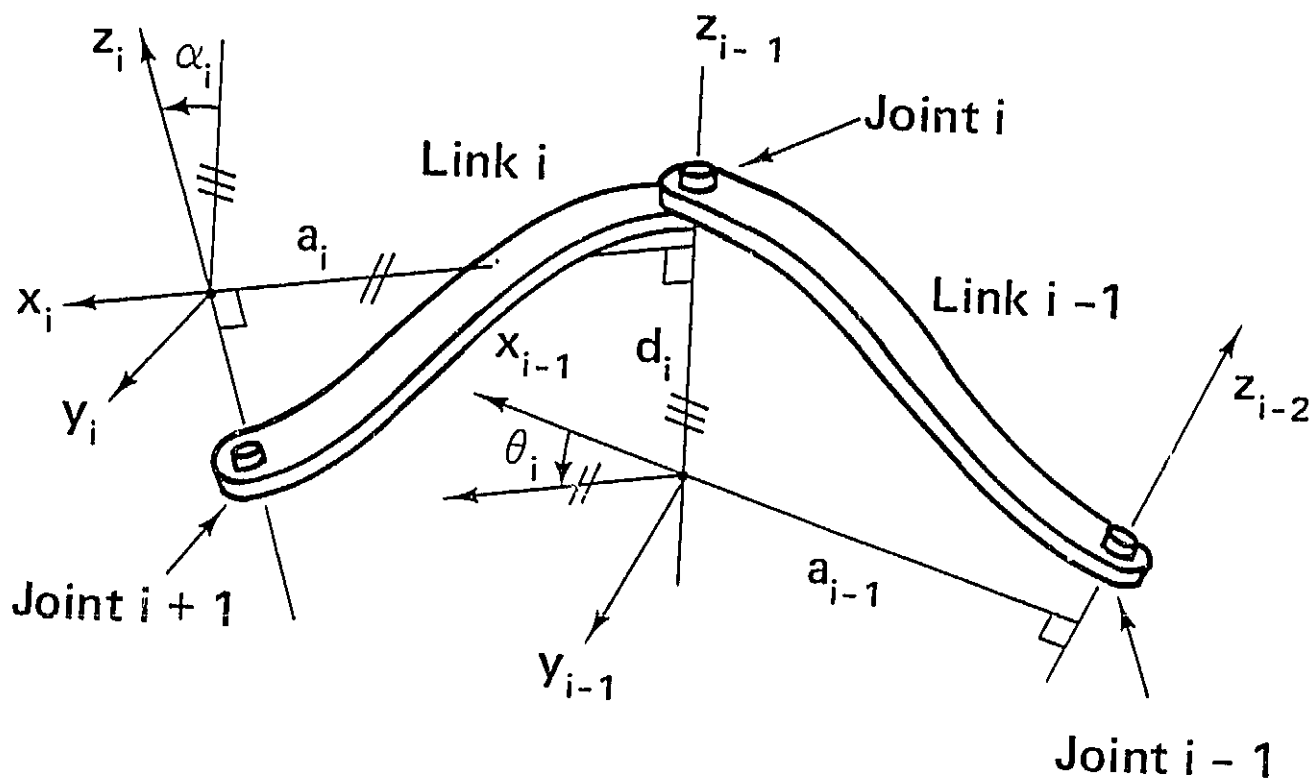


Figure 3.2.1-1 Link Coordinate System and Its Parameters

axis. If joint 1 is prismatic, then it is characterized by an angle parameter  $\theta_1$  and a joint variable  $d_1$ .

Once the link coordinate frames have been established for a robot arm, the kinematic configuration of the robot arm is related to the four geometric parameters for each joint. Based on the motion of the joints and the geometric parameters, robot arms can be classified according to the sequence of the joints from the base to the hand and their non-zero geometric parameters.

The advantages and disadvantages of four types of current robot arms are listed in Figures 3.1.2-1 through 3.1.2-4. Most of the existing industrial robots fall in one of the four classes. These four types of robot arms exhibit different workspaces (envelopes) which result in various accuracies at the end-point of the robot arm. Although the articulated robot exhibits poorest accuracy among these four types of robot arms, the space station application, (due to the master-slave operation in the supervisory control mode) may need an articulated robot arm. The arm operation would be similar to a human arm and may prove to have more advantages than other types of robots in terms of control and coupling. Most existing control algorithms direct a manipulator in the joint-variable space to follow a joint-interpolated trajectory. In some applications, there is a need for controlling the hand motion directly along axes relevant to the task environment. Such motion control is known as resolved motion control. Resolved motion means that the motions of the various joint motors are combined and resolved into separately controllable hand motions along the world coordinate axes. This implies that several joint motors must run simultaneously at different time-varying rates in order to achieve variable hand motion speed along any one world coordinate axis. This enables the user to specify the direction and speed along any arbitrarily oriented straight line for the manipulator to follow. This motion control greatly simplifies the specification of the sequence of motion for completing a task because users are usually more adapted to the Cartesian coordinate system than the manipulator's joint angle coordinates. Details about the resolved motion control algorithms are discussed in the control section.

### 3.2.1.2 Robot Arm Dynamics

The dynamic performance of robot arms is directly dependent on the efficiency of the control schemes/algorithms and the dynamic models of the arm. Improving the efficiency of these algorithms and obtaining better dynamic models of the arms are an integral part of robot arm control. The control problem requires the determination of the torque to be generated at each joint actuator for each set point on a precomputed arm trajectory in real-time. Though the control problem may be stated in this rather simple manner, the solution for arm control is complicated by the arm's inertial forces, and the coupling reaction forces arising from moving objects in the space station construction tasks.

The a priori information needed for manipulator control analysis is a set of closed form differential equations describing the dynamic behavior of the manipulator. Various approaches are available to formulate the robot arm dynamics, such as the Lagrange-Euler (L-E) [Bej74, Pau72], the Newton-Euler (N-E) [Arm79, LWP80a, Wa082], the Recursive Lagrangian (R-L) [Hol80], and the Lagrange form of Generalized D'Alembert (G-D) [LLN83].

The derivation of the dynamic model of a manipulator based on the Lagrange-Euler method is simple and systematic. The resulting dynamic equations of motion, excluding the backlash and the gear friction, are a set of second order coupled nonlinear differential equations. Bejczy [Bej74], based on the  $4 \times 4$  homogeneous transformation matrix representation of kinematic chain and the Lagrangian formulation, has shown that the dynamic equations of motion for a six-joint manipulator (Stanford arm) are highly nonlinear and consist of inertia loading, coupling reaction forces between joints (Coriolis and Centrifugal) and local force field loading effects. It has been recognized that the dynamic equations of motion as formulated are computationally inefficient [Pau72, TML80], and real-time control based on the "complete" dynamic model has been found difficult to achieve if not impossible [Pau72, LCM82]. For space robots, it is desirable to move the robots at reduced speeds to minimize the coupling inertial forces. Also, since there is no gravity loading effect on the links, simplified sets of equations can be used to model the dynamic behaviour of the space construction robots. In general, these approximate models will simplify the underlying physics by neglecting the Coriolis and centrifugal reaction terms [Bej74]. With these simplified models, the computation of the applied torques to the joint actuators is greatly reduced.

An approach which has the advantage of both speed and accuracy was based on the Newton-Euler vector formulation [LWP80a]. The derivation is simple, but messy, and involves vector cross-product terms. The resulting dynamic equations, excluding the dynamics of the control device and the gear friction, are a set of forward and backward recursive equations. This set of recursive equations can be applied to the robot links sequentially. The forward recursion propagates kinematics information (such as angular velocities, angular accelerations, linear accelerations, total forces and moments exerted at the center of mass of each link) from the base reference frame (inertial frame) to the end-effector. The backward recursion propagates the forces and moments exerted on each link from the end-effector of the manipulator to the base reference frame. Because of the nature of the formulation and the method of systematically computing the torques, computations are much simpler which makes it possible to achieve a short computing time.

The inefficiency of the equations of motion as formulated by the Lagrange-Euler method comes mainly from the  $4 \times 4$  homogeneous transformation matrices. To improve the computation time, Hollerback [HO180] has shown the recursive nature of the Lagrangian formulations. However, the recursive equations destroy the "structure" of the dynamic model which is useful to provide

insight for the design of controller. For control purposes, one would like to obtain a set of closed form differential equations (state equations) so that coupling reaction forces can be easily identified and appropriate controller can be designed to compensate their effects. In order to obtain an efficient set of closed form equations of motion, Lee [LLN83], based on the generalized D'Alembert principle, was able to utilize the vector and rotation matrix representation to describe each link's kinematic information, obtain the kinematic and potential energies of the robot arm to form the Lagrangian function and apply the Lagrange-Euler formulation to obtain the equations of motion. The generalized D'Alembert equations of motion enable a user to identify the following forces in functional form:

- (i) inertial forces/torques due to the rotational and translational effects of the links.
- (ii) velocity-generated (Centrifugal, Gyroscopic and Coriolis) reaction torques/forces between joints due to the rotational and translational link effects.

Three different formulations for robot arm dynamics have been discussed. The L-E approach is well structured and can be expressed in matrix notation, but computationally it is impossible to utilize for real time control purposes unless the equations of motion are simplified. The N-E method results in a much more efficient set of recursive equations, but they are very difficult to use for deriving advanced control laws. The generalized D'Alembert equations of motion give fairly well "structured" equations at the expense of higher computations. To briefly summarize the results, a user is able to choose between a formulation which is highly structured but computationally inefficient (L-E), a formulation which has efficient computations at the expense of the "structure" of the equations of motion (N-E) and a formulation which retains the "structure" of the problem with only a moderate computing penalty (Generalized D'Alembert). Table 3.2.1-1 shows the comparison of computational complexity among these three different approaches for robotic equations of motion.

### 3.2.1.3 Robot Motion Control Methods

Given the manipulator equations of motion, the purpose of robot arm control is to maintain a prescribed motion for the arm along a desired arm trajectory by applying corrective compensation torques to the actuators to adjust for any deviations of the arm from the trajectory. The movement of a robot arm is usually performed in two distinct control phases as previously discussed for the space application tasks. The first is the gross motion control in which the arm moves from an initial position/orientation to the vicinity of the desired target position/orientation along a pre-planned trajectory. The second is the fine motion control in which the end-effector of the arm dynamically interacts with the object using sensory feedback information from the external sensor(s) to complete the task.

| Approach                  | Lagrange-Euler  | Newton-Euler                           | Generalized d'Alembert  |
|---------------------------|---|--|---|
| Multiplications           | $\frac{128}{3} n^4 + \frac{512}{3} n^3 + \frac{739}{3} n^2 + \frac{160}{3} n$ | 132n                                   | $\frac{13}{6} n^3 + \frac{105}{2} n^2 + \frac{268}{3} n + 69$ |
| Additions                 | $\frac{98}{3} n^4 + \frac{781}{6} n^3 + \frac{559}{3} n^2 + \frac{245}{6} n$  | 111n - 4                               | $\frac{4}{3} n^3 + 44 n^2 + \frac{146}{3} n + 45$             |
| Kinematics Representation | 4 x 4 Homogeneous Matrices  | Rotation Matrices and Position Vectors | Rotation Matrices and Position Vectors                        |
| Equations of Motion       | Closed-form Differential Equations  | Recursive Equations                    | Closed-form Differential Equations                            |

where n - number of degrees-of-freedom of the robot arm

Table 3.2.1-1 Comparison of Robot Arm Dynamics Computational Complexities

Another way of viewing the operation of robot control is to divide the control into a motion planning phase and a servo control phase. Almost all current industrial robots perform motion planning by dividing the robot trajectory into phases. The first and third are an acceleration/deceleration, respectively, at a constant level. The second is a constant velocity section. The current industrial approach to robot arm control system design treats each joint of the robot arm as a simple servomechanism while the use of sensor driven motion (other than position and rate sensors) is only just beginning to appear in industry.

The current approaches reviewed above are inadequate for a number of reasons. The motion planning ignores the total torque capabilities of the robot over most of the motion cycle. It has recently been shown that the trajectory time can be reduced by a factor of between two or more through the use of more sophisticated planning and control techniques. The simple servo modeling currently used is inadequate because it neglects the motion and configuration of the whole arm mechanism. These changes in the parameters of the controlled system are significant enough to render conventional feedback control strategies ineffective. The result is reduced servo response speed and damping, which limits the precision, accuracy, stability and speed of the end-effector. Any significant performance gain in these and other areas of robot control require the consideration of elaborate dynamic models and sophisticated control techniques, and the exploitation of computer architecture.

For a multi-joint robot arm, development of suitable feedback control presents a substantial challenge, since the control problem is nonlinear and multivariable in nature. Several control algorithms which utilize the dynamic models discussed in the above section have been proposed to control industrial robots. Among these are: (i) computed torque technique [Pau72, Bej74, Lee82, Lee83, LCM82], (ii) the resolved motion rate control (RMRC) [Whi69], (iii) the resolved motion acceleration control (RAC) [LWP80b], (iv) the resolved motion adaptive control (RMAC) [LeL83], (v) the Cerebellar Model Articulation Controller (CMAC) [Alb75], (vi) the near-minimum-time control [KaB71], (vii) the nonlinear feedback control [SaL79], and (viii) the model reference adaptive control (MRAC) [DuD79] and other adaptive controls [LeC82b, KoG83]. Each of these techniques is briefly summarized in the paragraphs that follow.

### Computed Torque Technique

One of the basic control schemes is the computed torque technique [Mar73, Bej74] based on the Lagrange-Euler or the Newton-Euler equations of motion. Basically the computed torque technique is a feedforward control and has a feedforward and a feedback components. The feedforward component compensates the interaction forces among all the various joints and the feedback component computes the necessary correction torques to compensate any deviations from the desired trajectory. It assumes that one can accurately compute the counterparts of the inertial reaction forces, the Coriolis and centrifugal reaction forces in the dynamic



equations of motion to minimize their nonlinear effects, and can use a position plus derivative control to servo the joint motors.

### Resolved Motion Rate Control

In some applications, there is a need for controlling the hand motion directly along axes relevant to the task environment using resolved motion control. This enables the user to specify the direction and speed along any arbitrarily oriented straight line for the manipulator to follow. This motion control greatly simplifies the specification of the sequence of motion for completing a task because users are usually more adapted to the Cartesian coordinate system than the manipulator's joint variable coordinates. Whitney [Whi69] details the mathematics involved for using the resolved motion rate control in his paper.

### Resolved Motion Acceleration Control

The resolved acceleration control extends the concept of resolved motion rate control to include acceleration control. It presents an alternative position control which deals directly with the position and orientation of the hand of a manipulator. It assumes that the desired positions, velocities and accelerations of a preplanned hand motion are specified by the user. All the feedback control is done at the hand level. Details about this control method can be found in the third paper written by Luh et al [LWP80].

### Variable Structure Control

Young [You78] indicated that the theory of variable structure system (VSS) can be used to design a variable structure control (VSC) for multi-joint manipulators. The variable structure systems are a class of systems with discontinuous feedback control. The main feature of VSS is that it has the sliding mode on the switching surface. While in the sliding mode, the system remains insensitive to parameter variations and disturbances and its trajectories lie in the switching surface. It is this insensitivity property of VSS that enables the designers to eliminate the interactions among the joints of the manipulator. The sliding phenomena do not depend on the system parameter and have a stable property. Hence the theory of VSS can be used to design a variable structure controller which induces sliding mode in which lie the robot arm's trajectories. Such design of variable structure controller does not require accurate dynamic modelling of the manipulator; only the bounds of the model parameters are sufficient to construct the controller. The variable structure control differs from the time optimal control in that the variable structure controller induces sliding mode in which the trajectories of the system lie. Furthermore the system is insensitive to system parameter variations in the sliding mode.

### Cerebellar Model Articulation Controller

The CMAC is a table look-up control method which is based on neuro-physiological theory. It computes control functions by

referring to a table stored in the computer memory rather than by solution of analytic equations. It augments input commands and feedback variables into an input vector which is used to address a memory space where appropriate output servo controlled variable values are stored. All values representing n-dimensional angular positions, angular velocities and angular acceleration are stored. Since the generalized forces are functions of the controlled variables, these forces are obtained at no computational cost by indexing the table with the desired trajectory values. For practical applications several problems such as memory size management, computation cycle time, training requirements and accuracy need to be solved.

### Suboptimal Nonlinear Control Techniques

Other work, such as [HeC76] and [SaL79], incorporates the nonlinear terms in the model into the feedback loop of the control system to minimize the nonlinear effects of the system. Then a controller is constructed for the linear model. The use of nonlinear feedback components, to minimize the effects of the nonlinear coupling terms in a nonlinear control system and to transform the nonlinear system to a linear system that can be controlled using state feedback, is not new to control practitioners, but it is a good approach to control multi-joint robot arms. There is a substantial body of nonlinear control theory which may allow one to design a near-optimal control for mechanical manipulators. Hemami (1976) applied the nonlinear feedback control technique to a simple location system which has a particular class of nonlinearity (sine, cosine, and polynomial) and obtained decoupled subsystems, postural stability and desired periodic trajectories. His approach is different from the method of linear system decoupling where the system to be decoupled must be linear. Recently a suboptimal control design was developed by Lee and Chen [LeC83]. The feedback control law is composed of a nonlinear feedback for quasi-linearization and a suboptimal control law with switching function. The suboptimal control law is very similar to the optimal control law for linear quadratic problems. A switching function is applied for stabilization. The design also provides detour for obstacles. The total computational time for the control law is only slightly more than that of an open-loop control of a robot arm whose equations of motion are obtained from the Lagrange-Euler formulation.

Most of the existing control schemes discussed above control the arm at the hand or the joint level and emphasize nonlinear compensations of the interaction forces among the various joints. These control algorithms may be inadequate because they require accurate modeling of the arm dynamics and neglect the changes of the load in a task cycle. These changes in the payload of the controlled system may render the above feedback control strategies ineffective. Any significant performance gain in tracking the desired trajectory as closely as possible for all times over a wide range of manipulator motion and payloads require the consideration of adaptive control techniques as discussed below.

## Adaptive Control Techniques

Recently various adaptive control algorithms have been prepared. Among various adaptive control methods, the Model Referenced Adaptive Control (MRAC) is most widely used and relatively easy to implement. The concept of model referenced adaptive control is based on selecting an appropriate referenced model and an adaptation algorithm which modifies the feedback gains to the actuators of the actual system. The adaptation algorithm is driven by the errors between the referenced model outputs and the actual system outputs. Dubowsky [DuD79] proposed a model referenced adaptive control which uses a linear second-order time invariant differential equation as the referenced model for each degree of freedom of the robot arm. The manipulator is controlled by adjusting the position and velocity feedback gains to follow the model. A steepest descent method is used to update the feedback gains. Koivo [KoG83] proposed an adaptive self-tuning controller using an autoregressive model to fit the input-output data from the manipulator. A recursive least square estimation scheme is used to estimate the parameters in the model for updating the control gains. Both control algorithms assume that the interaction forces among the joints are negligible.

Lee [LeC82b] proposed another adaptive control to track a desired trajectory as closely as possible for all times over a wide range of manipulator motion and payloads. The proposed adaptive control differs from the above adaptive controls by taking all the interactions among the various joints into consideration. The adaptive control is based on the perturbation equations in the vicinity of a desired trajectory. The highly coupled nonlinear dynamic equations of a manipulator are linearized about a preplanned joint trajectory to obtain the perturbation equations. A feedback control law is then designed to control the linearized system about the desired trajectory. The controlled system is characterized by a feedforward component and a feedback component. Using the Newton-Euler equations of motion, the feedforward component computes the nominal torques which compensate all the interaction forces among the various joints. The feedback component computes the variational torques which reduce the position errors of the manipulator along a nominal trajectory. A recursive least square identification scheme is used to identify the system parameters of the linearized perturbation system. The parameters and the feedback gains of the linearized system are updated and adjusted in each sampling period successively to obtain the necessary control effort. This adaptive control strategy reduces the manipulator control problem from a nonlinear control to controlling a linear control system about a desired trajectory. A clear advantage of the formulation is that the nominal and variational torques can be computed separately and simultaneously. Computer simulation studies between this adaptive control and the control law obtained from the computed torque technique are tabulated in Table 3.2.1-2.

Recently Lee [LeL83] also proposed a resolved motion adaptive control which adopts the ideas of "Resolved Motion Rate Control" and "Resolved Motion Acceleration Control" and extends the

| Various Loading Conditions                    | Joint | PD Controller       |                     |                               | Adaptive Controller |                     |                               |
|---|-------|---------------------|---------------------|-------------------------------|---------------------|---------------------|-------------------------------|
|   |       | Max. Error (degree) | Trajectory Tracking |                               | Max. Error (degree) | Trajectory Tracking |                               |
|   |       |                     | Max. Error (mm)     | Final Position Error (degree) |                     | Max. Error (mm)     | Final Position Error (degree) |
| No load and 10% error in Inertia tensor       | 1     | 0.089               | 1.55                | 0.025                         | 0.020               | 0.34                | 0.000                         |
|   | 2     | 0.098               | 1.71                | 0.039                         | 0.020               | 0.36                | 0.004                         |
|   | 3     | 0.328               | 2.85                | 0.121                         | 0.032               | 0.28                | 0.002                         |
| 1/2 max. load and 10% error in inertia tensor | 1     | 0.121               | 2.11                | 0.054                         | 0.045               | 0.78                | 0.014                         |
|   | 2     | 0.147               | 2.57                | 0.078                         | 0.065               | 1.14                | 0.050                         |
|   | 3     | 0.480               | 4.19                | 0.245                         | 0.096               | 0.83                | 0.077                         |
| Max. load and 10% error in inertia tensor     | 1     | 0.145               | 2.53                | 0.082                         | 0.069               | 1.20                | 0.023                         |
|   | 2     | 0.185               | 3.23                | 0.113                         | 0.069               | 1.22                | 0.041                         |
|   | 3     | 0.607               | 5.30                | 0.360                         | 0.066               | 0.58                | 0.019                         |

Table 3.2.1-2 Comparisons of the PD and Adaptive Controllers

above adaptive control concept to control a manipulator in Cartesian coordinates for various loading conditions. The proposed adaptive control is performed at the hand level and is based on the linearized perturbation system along a desired hand trajectory. The control method assumes that the desired hand positions, velocities and accelerations along a path/trajectory in Cartesian coordinates are given. This resolved motion adaptive control is very suitable for space robots because one wants the manipulator hand to move in a known world coordinate axis in a coordinated rate and position control under varying loading conditions.

#### 3.2.1.4 Robot Control Research Directions

The research problem in closed-loop force feedback control is to find a control strategy which utilizes the force signals from a wrist sensor to appropriately servo the arm to track the desired force and position trajectories in completing the assembly tasks. In the fine motion phase, vision is obscured and the end effector of the arm is in contact with its target. Thus during this phase force feedback must be used to guide the arm in its operation. In particular, in batch assembly where high precision tasks such as fitting and insertion of mechanical parts are common, this final phase takes on added importance. Presentday industrial robots do not effectively use force sensory information to perform fine motion control for these high precision assembly tasks.

Past work in force control has been performed at various institutions using joint sensors, pedestal sensors and even torque sensing by monitoring the armature currents of the motors. The first sensor-controlled manipulator was demonstrated by Ernst [Ern62] at MIT in 1961. The computer-controlled mechanical hand MH-1 was equipped with tactile sensor which could "feel" blocks and stack them without assistance from the operator. In 1962 Tomovic and Boni [ToB62] developed a prototype hand equipped with a pressure sensor which sensed the object and supplied an input feedback signal to a motor to initiate one of the two grasp patterns. These basic sensor control schemes were heuristic in nature and the control algorithms for the hand were crude.

In 1973 Bolles [BoP73] demonstrated the assembly of a water pump by a computer-controlled Stanford arm using both visual and force feedback. Force feedback techniques together with a heuristic circular search were used successfully in locating the holes for assembly. The sensing of forces was done via monitoring the armature currents of the joint motors. Though the assembly of water pump was successful, the fraction of time spent to perform force feedback control was much too long. About the same time, Will [WiG75] and his associates at IBM developed a computer-controlled manipulator with touch and force sensors to perform mechanical assembly of a 20-part typewriter. Inoue [Ino74] at the MIT Artificial Intelligence Laboratory worked on the artificial intelligence aspect of force feedback. A landfall navigation search technique was used to perform initial positioning in a precise assembly task.

In 1976 Whitney [Whi76] presented a force feedback

strategy called accommodation. The method was simple and the strategy was embedded in the force feedback gain matrix (or compliance matrix). At the same time, Paul and Shimano [PaS76] extended the work at the Stanford Artificial Intelligence Laboratory by using compliance. Their analysis showed that the required computation period exceeded the desired sampling period therefore an "approximate" solution was devised for the compliance control. Shimano [Shi78] implemented the resulting compliance control method on the AL system. His control scheme was open-loop and consisted of motions controlled by external forces to comply along the desired joint coordinate axes. Later, he incorporated feedback into the control scheme. Nevins, et al [Gro72, Dra77] investigated the amount of information from the compliance of the environment. This work developed into the instrumentation of a passive compliance device called remote center compliance (RCC) which was attached to the end of the manipulator for close parts-mating assembly. The device defines a center of compliance at the tip of the peg. Because of the location of the center of compliance, insertion of pegs of different sizes requires different RCC's - an undesirable effect.

Lee [Lee80] formulated the interaction between the end-effector and the object as a finite state machine. This method again is still quite heuristic in nature. Raibert [RaC81] uses a hybrid control scheme for force control. Conceptually the technique seems promising but implementation information was lacking from the work and the experimental results were not clearly explained.

Almost all the force feedback control schemes work in conjunction with some heuristic search techniques, and heuristics usually lead to undesirably long assembly time. Moreover, these control strategies are limited in scope due to their inability to improve their performance based on past experience. Research effort must be directed toward designing an active compliance control that builds on these strategies and incorporates self-improvement by utilizing pattern recognition and learning techniques.

In addition to developing improved strategies that will allow faster and more accurate execution of the arm's fine motion control, research effort must also be directed toward developing the force control structure/architecture (or control hierarchy) to support the computations for these strategies in order to ensure the speed required for real-time control of the overall arm motion.

### 3.2.2 Sensing and Machine Vision for Space Robots

The use of sensors plays an important role in extending the capability of robots to a less structured (and eventually unstructured) environment. In general, the word "sensors" implies "external" sensors as opposed to internal sensors such as the potentiometer and tachometer (or encoder) embedded in the DC torque motor at each joint. Basically there are two main classes of external sensors: (a) Contact sensors and (b) Non-contact sensors.

Contact sensors sense force/torque, and touch/pressure while in physical contact with objects. Noncontact sensors sense visible (or non-visible) images, range, and the presence of objects without making any physical contact.

In general, the sensors are mainly used for the following functions:

- (1) Obtaining a priori information about the workspace and workpieces.
- (2) Correction of errors in the robot arm joint position and its end-effector.
- (3) Detection of potential faults (such as obstacles) and minimization of their effects.
- (4) Automatic inspection of the end-products for possible assembly faults.

In this section, we briefly describe different types of sensors that are commercially available and sensors development in research laboratories. It is expected that later evolutionary versions of these sensors may be utilized in the proposed space robotics systems.

#### 3.2.2.1 Contact Sensors

Contact sensors include force/torque sensors, tactile sensors, and pressure sensors. Only the force and tactile sensors will be described in more detail.

##### Force/Torque Sensors

Force/torque sensors are used primarily for measuring the three resolved orthogonal components of force and torque at the assembly interface. Three types of force/torque sensors can be used for this purpose:

- (1) A joint force sensor measuring the Cartesian components of force and torque acting on the robot joints and added vectorially. For a joint driven by a DC motor, sensing is done by measuring the armature current. This method requires no special transducer, but is relatively inaccurate because the measured forces include those that are not transmitted to the other joints and the end-effector, such as joint friction.
- (2) A wrist force sensor tailored to be mounted between the last joint's mounting plate of the robot and its tool. It consists of strain gauges that measure the deflections of the mechanical structure due to external force exerted on it. The wrist sensors are sensitive, lightweight and relatively compact in design.

- (3) A pedestal force sensor that provides a base for assembly operation and measures the components of force and torque applied to a workpiece on the pedestal. It also consists of strain gauge transducers that measure the deflection of the mechanical structure. Drake [Dra77] built such a device for testing, but it did not have sufficient resolution for some assembly operations.

### Tactile Sensors

Tactile sensors are used to obtain information associated with the contacts between the fingers of a manipulator hand and objects in the workspace. They are normally much lighter than the hand, are sensitive to forces much smaller than those sensed by the wrist force sensors, and are usually mounted on the inner surfaces of one or both fingers. These sensors may be used to obtain information about a workpiece before it is acquired and about the location of grasping points as well as any workpiece slippage during acquisition.

Tactile sensors may be classified into two classes: binary and analog. A binary tactile sensor can be realized by micro switches. Garrison built a gripper with 100 pneumatic binary tactile sensors located on a grid. The sensors consisted of contact terminals, a thin metal sheet with elastic shallow spherical domes, and a flexible insulating rubber sheet on the outside. Physical contact was sensed whenever external pressure exceeded a preset threshold, causing activation of a snap-action switch consisting of a dome and a terminal.

An analog tactile sensor is a compliant device whose output is proportional to the force or pressure exerted on it. Among the uses for such a sensor is the measurement of gripping forces and eliciting information about the object between the fingers. Hill [Hil77] built a manipulator hand with a wrist force sensor and analog tactile sensors. Seven outer sensing plates and a matrix of 3 by 6 inner tactile sensors are mounted on each finger. The force on each sensor acts against a compliant washer, displacing a vane that controls the amount of light received by a phototransistor from a light-emitting diode.

LORD Corporation has recently introduced an 8 x 12 tactile sensor with eight bits of pressure resolution and 0.1 inch spacing. Integrated circuit tactile sensor without chip circuitry are currently under development at Carnegie Mellon and the University of Michigan. From a study by Harmon [Ha83] it is clear that tactile sensing will play an important role in future assembly systems.

#### 3.2.2.2 Noncontact Sensors

Noncontact sensors are useful in identifying and finding the location of the parts in sensor-controlled manipulation and for inspection. The major categories of noncontact sensors that have



been used with robot systems are electro-optical imaging sensors, proximity sensors, and range-imaging sensors.

Electro-optical imaging sensors provide the most commonly used "eyes" for industrial robots and visual inspection. A wide variety of devices are available including several types of one and two dimensional cameras. Further, lighting is used in several ways: backlighting, line striping, shuttered, studio, unstructured, etc. Solid-state TV cameras, interfaced with a computer, however, probably constitute the least expensive and most easily available imaging sensors. These cameras scan a scene, measure the reflected light intensities within a raster of, say, 512 x 512 x 8 bit pixels, convert these intensity values to analog electrical signals, and feed this stream of information into a computer within 1/30 of a second. These signals may either be stored in the computer's memory for subsequent processing or they can be pre-processed in real time, with a consequent reduction in the memory requirements.

The visible imaging area is developing rapidly, partly in view of the large potential market for home video recorders and their possible replacement of film-based still and movie cameras. Impressive 512 x 512-element color imagers have been reported, and an imager capable of recording at up to 2000 frames per second over 248 x 192 elements has recently been described. Unidimensional solid-state cameras, using linear-diode arrays that vary from 16 to 1872 elements, are also available commercially. These devices can perform a single linear scan and are particularly useful for sensing objects that are in motion relative to the camera, such as workpieces moving on a conveyor belt.

### 3.2.2.3 Machine Vision

This section describes the current research and capabilities of the machine vision area based upon image data obtained from the sensors described above. While most of the work described below is based on data derived from visible light sensors, it can be extended to other types of "illumination" and sensors. This section will concentrate more on the algorithms of current vision techniques and less on the sensors.

#### 3.2.2.3.1 Binary Vision and Gray Scale Vision

Most of currently available systems for machine vision use binary vision techniques and are used only in the inspection area. These systems typically use a solid-state sensor with small spatial resolution (128 x 128 or 256 x 256). The SRI Vision Module is prototypical of these types of systems. These systems use recognition algorithms based on statistical features and some relatively simple geometrical features. Many of the algorithms use template matching techniques. These systems use simple computer architectures but are capable of performing the inspection tasks in relevant time however they have serious drawbacks in that most of the systems require special lighting to back light the part.

Binary vision systems have been used in assembly

operations. However, it is felt that these types of systems are not adequate for assembly operations in general due to the special conditions needed for the system. It is believed that due to the limitation placed on these types of system (with respect to lighting and general application) that binary vision will not be useful in space robots. It will be necessary to use gray level information as desired from the sensors discussed above (e.g. 8 bits/pixel).

A typical gray scale vision system will consist of a preprocessing step to clear or enhance the image derived from the sensor (e.g. medium filtering to remove salt-and-pepper noise), a segmentation step where "important regions" in the image are found (i.e. "objects" are segmented from the background in scene), and a feature extraction step where salient features of the segmented regions are measured (e.g. object area, object perimeter, and other shape measures). The final step is classification or object(s) recognition with respect to type and orientation. The classification step can be comprised of statistical, syntactic or other approaches. Many knowledge-based systems combine several of the steps above into a slightly different approach.

#### 3.2.2.3.2 Machine Vision Subtask Areas

The areas discussed below can generally be placed in the segmentation/feature, extraction/classification levels of the overall problem. Before we discuss some specific approaches and research areas in machine vision it should be emphasized that these topics represent "individual" areas or subtasks in the overall vision problem and that such a problem as "inspect the bolt" may require the integration of several topic areas to accomplish the given task. To date every overall machine vision system whether it be for inspection or assembly has been a "special" design of a group of analysis algorithms. Machine vision research can be characterized into specialized areas. For example, researchers are working in very special applications such as turbine blade inspection or working on special topics such as occluded parts recognition. More work needs to be done to integrate this "special topics" research into overall systems. Below we shall describe some of the current important research topic areas.

##### Model-Based Vision

In most cases in inspection and assembly operations the parts and object are known completely a priori and are probably stored in a CAD/CAM system or at least one knows something about the objects. In many cases one can then construct a relevant mathematical model of the object that can be used in the recognition/orientation problem. In many cases this model can be a description of the shape of the object as seen by the sensor and as derived by the output of the segmentation process. Due to imperfect segmentation one never obtains a "truly segmented" object. By using a model for the part one can often recognize it after very poor segmentation.

The use of information from CAD/CAM data bases have

aided in the design of these object models. Most object models used have been based on shape information.

### Occluded Parts

As mentioned above one does not, in general, obtain adequate segmentation of objects due to the segmentation algorithms and/or parts or other objects blocking the sensor's field of view. Various techniques are currently being used to approach this problem. The model-based approach has been used along with template matching techniques to recognize occluded parts. These techniques generally use shape information derived from whatever segmentation operation is performed.

Another approach uses range information (discussed below) to find which part is in front (or in back) of another part. This information is then used to identify the object. If the object or group of objects can be gripped and picked up then various other techniques may be used to separate parts.

### Three-Dimensional (Range) Information

The extraction of depth (or range) information has been investigated since the beginning of interest in machine vision. Various techniques are currently of interest. The first technique is stereo ranging from multiple sensors. This technique has problems in matching parts of the scene as viewed in each sensor. This so-called correspondence problem is very difficult and various approaches can be used to solve it. In some cases, one sensor can be used to obtain depth information if the desired scene is viewed from different perspectives.

The second approach is that of laser (or radar) ranging. The scene is scanned by one of these devices and depth information is obtained. It is possible to achieve a very accurate result using these techniques (far more accurate than stereo techniques).

The last approach is that of structured lighting whereby a structured pattern of light is focused on the scene and depth information is obtained by observing how the pattern of light is deformed by the scene. The disadvantage here is that only a binary image is obtained.

An important question is how does one use the depth information after it is obtained. It should be noted that in many ways each of the three approaches above produce different types of range information. Even if an ideal range map is obtained one must be careful in using it. Range information has been used in occluded part recognition, three dimensional shape analysis, and other recognition problems. The use of range information has not been fully exploited.

### Shape Information

In many applications the shape of an object is a very powerful measure used in identifying the object and its

orientation. Shape measures or shape descriptors can be obtained using various techniques. These include Fourier methods, splines, shape-from-shading, and CAD/CAM models. In many cases shape is the single most important feature in identifying an object.

### Dynamic Scene Analysis

This area involves the use of temporally redundant information available in a sequence of images taken of a scene. These images can be obtained because objects in the scene move or because the sensor moves. Areas of research here include depth extraction, shape change analysis, segmentation, edge detection, object tracking, and guidance.

The above overviews some of current areas of interest in machine vision. We believe that all of these areas have application to spaceborne robotics.

### 3.2.3 Robot Languages, Computers, and Data Communications

#### 3.2.3.1 Robot Languages

Robot Languages (RLs) can be classified into two general categories: teach by doing and text-oriented programming. The former is simple and easy since it is done by guiding a robot with a joystick or a teach pendant, and recording a number of points in the path. The recorded information can later be replayed. This is currently in wide use because of its simplicity and practicality. However, the teaching method encounters severe difficulties when sensors are added to the systems.

The second method is based on text programs similar to general programming. This requires complex compilers/interpreters, debugging facilities, interface to intelligent environments (world modeling system, CAD/CAM database, graphics aid, etc.). This record programming method has great flexibility and is likely to be essential for incorporating sensors into the robot system. However, it is very difficult to design the required languages. The development of off-line text oriented languages is a subject of current and future research. Note that this method can only be made possible with the corresponding development of expertise in many different disciplines. Application of newly emerging artificial intelligence concepts should play a deciding role. Meanwhile, a hybrid combination of the two programming methods can be regarded as a viable solution to most space robotic applications.

It is important to understand the current status of RL developments and then determine research directions. To this end we carried out an extensive, comparative study of 14 existing RLs in use or under development [BoS82]. Given below is a brief description of some of the fourteen languages studied as well as some new ones recently developed (see [BoS82] for detail). The following is a brief survey of a few of the more important current languages:

AL, developed by the Stanford Artificial Intelligence Laboratory, is a high-level RL with ALGOL-like control and block structure; data types of scalars, vectors, rotations, frames and transforms [MuG79]. It also has mechanisms for dealing with force sensors and maintaining the connectivity of an assembly. AL is not yet widely used because it was developed on specialized computing facilities and the sophistication is beyond most current robot users.

IBM has been an industrial leader in RL development, beginning with their early languages ML and EMILY. IBM also proposed a task-level language called AUTOPASS which is oriented towards objects and assembly operations with English-like syntax [LiW77]. More recently, they have developed and used AML which is a high-level interactive language with: interpretive nature; four data types of integer, real, string, and aggregate; control constructs similar to PASCAL; subroutine oriented features and interactive debugging facilities [IBM81]. One of its most significant features is the ability to handle sensor information and the inclusion of exception-like handling of their inputs. It is, however, heavily dependent upon unique vendor equipment.

RAIL is Automatix's language to control their robotic arc welding, assembly and inspection systems. Its data types include integers, real numbers, character strings, logical data, arrays, points, paths, and reference frames. Although it contains structured control constructs (e.g. IF--THEN--ELSE, WHILE, etc.), it requires an explicit expression of each motion segment [Aut82a], [Aut82b]. Particularly, a smooth transition between a series of paths having different speed schedules can be achieved by listing a series of paths with one MOVE command.

General Electric has developed a high level procedural language called HELP which is relatively easy to learn; supports the structured programming and simultaneous arm movement and has a special set of built-in functions/subroutines for robot operations [GE82]. MCL, an extension of numerical control language, APT, by McDonnell Douglas Corporation, is a high-level language for the off-line programming of industrial robots and associated equipment [Bau81], [Old81], [McD82]. Purdue University developed a language called PAL in which tasks are represented in terms of structured cartesian coordinates and every motion statement is described by homogeneous transformations [Tak81]. Shin et. al. developed an explicit language, RCL, to program a sequence of steps to accomplish a robot task [SVM82]. Emphasis was placed on the capability of low level control and/or sensing experiments for research purpose. RPL, a FORTRAN-like user language at SRI International, is designed to facilitate the writing and debugging of application programs for material handling, inspection, and assembly [Par81]. SIGLA, designed for the Super-Sigma robot, runs using only 8K bytes of memory and still provides features such as parallel task control and variable instruction sets for software tailoring [Sal78]. Finally, Unimation has developed an explicit language, VAL, for their PUMA robots [Uni79].

Current research in robot languages is directed

principally in three directions:

- o automatic development of robot program
- o development of object level program
- o the use of CAD information to assist in robot program development.

All three of these areas represent relatively new efforts. Latinbe [Lat83] and his associates have developed a system for automatically programming insertion operations using force feedback. Ambler [Amb83] is developing an object level programming system in which one describes action in terms of the objects to be manipulated and Volz, Woo and Langier [WWV82, Lan83] are developing techniques for automatically determining gripping position and other object features from a CAD database.

### 3.2.3.2 Computers

Although none of the proposed Space Station application robot computers currently exist, it is possible to analyze the current capabilities and define probable research directions on the basis of current capabilities for both ground and space robotic computers built or proposed to date.

Earth based robot computers can be classified into two different categories. One is the use of traditional general-purpose uniprocessors (of mini size), treating robots and their associated sensors and equipment as I/O devices. The other class involves the new development of dedicated computers for robotic systems. The former does not have anything different from typical commercial minicomputers (i.e. reasonable reliability, availability, components with reliability of commercial grade, etc.). The main advantage associated with this first category is the high level of experience with these known computers, and hence users do not require significant additional training. Further they are readily available and have a known, predictable performance and reliability history.

The second class of dedicated robot computers are just now emerging into system use. Typically, they are constructed using off-the-shelf microprocessors, memories, etc. The implementation naturally tends toward use of multiple computers or multiprocessors. The chief motivation for this category is that (i) robot systems consist of well-defined subsystems (e.g. control, sensing, etc.) with weak interactions between the subsystems, and (ii) the advances in VLSI technology have made available off-the-shelf components with great computation power at inexpensive costs. Notice, however, this category has mainly relied on the use of general-purpose, off-the-shelf components. It is believed this trend will continue to grow in importance in the future. Particularly, research efforts will focus on interconnection, intercommunications, task partitioning, distributed operating system, programming languages, I/O interfaces, synchronization and the like.

Concurrently with the above developments, special chips

for robotic applications are beginning to receive considerable attention. For example, calculations of inverse Jacobian matrices are required for robot control. The calculation has to deal with structured data and hence can be mapped into a dedicated chip. Likewise, special chips can be fabricated for processing robot vision data. This tendency will continue to expand rapidly as our understanding of robot computations progresses. Two fundamental research subjects to consider are: (1) transformation of robot computations into chips, and (2) the design of high speed but inexpensive communications among special purpose chips. The former is concerned with typical chip fabrication and characterization of robot computations. The latter is an architectural issue which is now seen as the major obstacle to overcome.

As an example of space computers, the Jet Propulsion Laboratory (JPL) designed and breadboarded a Self-Test and Repair (STAR) computer. STAR primarily used hardware-subsystem fault-tolerant techniques, such as functional unit redundancy, voting, power-spare switching, coding, and self-checks. Task-level rollback was also incorporated in the design which represented the most advanced fault-tolerant techniques in the 1960's. [AVI71]. Another fault-tolerant uniprocessor designed as a satellite computer is the Fault Tolerant Spaceborne Computer (FTSC) [FTSC76]. It is a 2-bit, general-purpose computer with a 60K-word memory and 5 micro-second average instruction execution time. Error-detection/correction codes and bit-sliced sparing are extensively used to tolerate failures.

With the advent of microprocessors, multiprocessor/multicomputer based space computers are beginning to appear. The Fault Tolerant Building Block Computer (FTBBC) is an experimental set of VLSI chips that allow construction of reliable multiprocessors with off-the-shelf chips. [Rennels80].

Though slightly different from most space computers, Software Implemented Fault Tolerant Multiprocessor (FTMP), designed by C.S. Draper Laboratories, are intended for real-time control of aircraft. These experimental systems are designed for a failure probability of  $10^{-9}$  during a 10-hour mission.

To sum up, space computers must generally emphasize reliability whereas current earth-based robot computers normally are designed for computational capacity without stressing reliability. Since space robot computers must have both characteristics the future research direction should be to develop multiprocessors or multi-computers with these two characteristics.

### 3.2.3.3 Computer Communication Networks

While communication between computers has been an essential part of major computer installations for many years there is still a great deal of activity in the development of more sophisticated communication technology particularly in high speed networks. Within the past three years there has been significant progress in the area of computer network standards and by the time of the projected space station project one can expect the seven

layer ISO communication model to be fully specified. Communication in space will be able to use many of the concepts developed for land based systems, but will require special attention in some areas because of the high data rate required and physical arrangement of the equipment.

There are three areas of concern in developing the data communications network for the space robotics activity:

- o topology
- o physical interconnection
- o communication protocol

There are six basic logical topologies which might reasonably be considered for the space robotics communications network system: point to point, star, hierarchial, general network, bus and ring. Each can be characterized by several key factors as the total number of nodes( $N$ ) in the network increases. These factors include average message delay, message traffic density, total connection costs, and the number of connections per node. Additional factors in comparing topologies include ease of routing messages between nodes and reliability in recovering from single point failures. Of course, comparison of different topologies requires a statistical knowledge on the message distribution in the network (e.g. uniform or exponential distributions) and also some fundamental nature of connection networks (e.g. uni- or bi-directional).

Completely connected networks have a dedicated link between each pair of nodes. Since it requires  $N \times (N-1)/2$  paths for  $N$  nodes, (i) connection costs grow as  $O(N^2)$ , and (ii) the network rapidly becomes under-utilized (i.e. message density in the network decreases rapidly with the increase of  $N$ ). Since there will be only a relatively small number of processors in the system, however, one might still consider using pairwise point to point communication between those processors which need to communicate. This would simplify some of the protocol issues; no target or source address information would be required and connection establishment procedures would not be extensive. However, these low level protocols are already being effectively and inexpensively implemented in hardware. By the time of implementation of the proposed space station construction project, these issues should be of no consequence.

Star configuration and hierarchial configurations also use a set of pairwise point to point links, but reduce the number of links by putting more intelligence for routing and message passing in the individual nodes. Two drawbacks are that (i) a single failure at the center node disable the entire network, and (ii) the center node could become a bottleneck slowing down message traffic. The latter is particularly important in view of the high data rates expected and the requirement that any pair of processors be able to communicate. The general network topology requires a general facility for routing, store and forward (or circuit establishment) and is probably more complex than required for the projected applications.



The remaining two topologies, bus and ring, both effectively allow any attached unit to communicate with any other attached unit. They differ in how they use the communication media and in their low level protocols they use. The ring network relays messages from node to node one-way around a loop. (It is easy to see an extension of this to include an additional loop forming double loops for two-way relays.) The ring is limited to a relatively small number of computers because average message delay and message traffic density increase linearly with the number of nodes on the ring. There is also concern because failure of a single node or path segment in the ring may disable the communication net. Nevertheless, techniques have been proposed for recovery of network in case of failure.

Time-shared bus systems are similarly limited to a relatively small number of nodes mainly because the time-shared bus may become a bottleneck as the number of nodes increases. Also, there are end to end distance limitations due to propagation delays. If only a single bus is used and it fails, then the entire system will collapse. Use of multiple buses may solve the problem of bottlenecking and reliability but increases link and switch costs on the order of  $N^2$  if there are  $N$  buses for  $N$  nodes. Both ring and bus networks are the subject of extensive research and development today and significant new products are appearing on the market. Data rates in the range of  $10^6$  to  $5 \times 10^7$  bits per second are being achieved.

The physical interconnection mechanism is separate, though related to the logical interconnection. Indeed, a logical pairwise point to point network can be implemented on a single bus like cable system by using frequency division multiplexing. Three types of physical data transmission media might be considered:

- o some type of cable (e.g. coaxial, fiber optic)
- o electromagnetic radio transmission
- o modulated light transmission

The first would most nearly match the high speed data communication systems being developed today. All of the high speed bus structured communication systems, such as Ethernet, NET1 and Wangnet and ring structures such as the Apollo ring use cable as the transmission media. Some other systems do make use of microwave or laser transmission, but only as a point to point extension or interconnection between two cable systems. Technically any one of several such systems which will certainly be in existence by the time of the projected flight should work acceptably well. However, the cable would present a floppy tether to the robots. This would cause several categories of problems to the operation of the robots.

First, the cable would be floating partially loose in space. Operation of a robot would then have to consider the possibility of the cable floating into the workspace at any time. Maneuvering of the robots would have to be done very carefully so as to not entangle the cable. Moreover, at times the cable might provide a disturbance to the position and attitude control system

of the robot. Consequently, a non cable transmission media is preferred.

Relatively little has been done to date in the way of radio or optic communication networks. A radio packet switched network has been implemented experimentally in the San Francisco Bay area. However, its data rate was relatively low. While some of the basic techniques being developed for today's high speed bus and ring based data communication networks are amenable to radio or light based transmission and sufficiently high data rates are possible with these transmissions, it is unlikely that they will be developed in a suitable form in the commercial marketplace. Also, while radio or laser transmission systems may be likely in the next 10 years, they will be developed for transmission over longer distances. As such, propagation delays will be significantly larger than those to be encountered in the space station construction application and design parameters taking these delays in account will affect both some levels of protocol involved and the performance that may be expected in ways detrimental to the intended application.

The most significant aspects of current research direction are in the area of bus and ring style high speed factory communication networks. Many of the significant problems are in the protocol and software area. Their solution should carry over to the space application. The primary work which will be needed for space applications will be in developing and incorporating the low level transmission mechanisms to be used.

### 3.2.4 Expert and Knowledge Based Systems

#### 3.2.4.1 Problem Solving

Distributed Problem Solving (DPS) has been one of the major research areas in artificial intelligence. Several techniques have been developed for designing programs that can solve novel problems. Most of the approaches for problem solving assumed only one agent and one database containing all information required for solving the problem. Many problems have two or more agents and the information required to solve the problem may be collected from several sources. The last few years have seen growing concern with distributed problem solving.

Distributed problem solving addresses situations in which there may be several agents, information required to solve the problem may be collected from disparate sources such that no single source has all the information required to solve the problem, and several processors may work on the same problem. Clearly, some issues of distributed processing become relevant in distributed problem solving, but the emphasis of research is on problem solving using incomplete information. As in the space robotics applications scenarios when there are several processors trying to solve a problem in this DPS mode some issues become particularly important. Two major issues are the extent of functional autonomy and representational diversity. If each agent is autonomous then the synchronization problems become trivial but the task of problem

solving using partial information becomes difficult. Which information to be communicated where and when is an important decision in most systems. The representation problem is important because the information is acquired by different agents in different form and hence is stored using different representations. The transformation of information from one representation to other must be performed for proper assimilation. Yet another problem, sometimes called the truth maintenance problem arises due to the fact that multiple agents are capable of affecting the environment. At a given time a problem solver must have correct information about the state of the system to be able to solve the problem.

DPS is being applied in many diverse fields. We are not aware of its application to multi-robot systems, however. In a space station assembly task, if it is desired that the system perform in an environment that is not completely controlled in terms of the location orientation and quality of components then DPS seems to be an appropriate approach. By using DPS the system will integrate information from sensors and robots to perform the required assembly task that will be stored in a knowledge base using procedural representation. The description of the assembly task in the knowledge base will give general steps, leaving specific situations to the problem solver.

Conventional robot control techniques are incapable of working in a novel situation. All motions of robots are designed in an a priori fixed environment. The path followed by a robot is fixed considering either an ad hoc technique or some optimization criterion. This path does not consider any new object or a new location of an object. A flexible system, such as the one to be used on a space substation will use motion planning considering a given configuration of objects to find a path to move an object from one location to other. Very little research has been done on this important problem.

Spatial reasoning is an important part of motion planning. Techniques based on known dimensions of objects to decide about the plausibility of a suitable path are being explored. These techniques consider dimensions of the object to be moved and find all possible paths that will allow the object to safely pass. In a complex situation, if no such path exists, the system may have to change the environment to make a path. In the Space Station assembler case, a request to the supervisor would be initiated for object removal.

#### 3.2.4.2 Knowledge Based and Expert Systems

The last few years have seen significant advances in expert system technology. Most expert systems are designed to interact with humans. In the present context of a space construction or servicing facility, the expert systems should have minimal interaction with humans. The required information should be acquired by the sensors. Only in exceptional cases should the human supervisor or the ground station be contacted.

Expert systems will be extensively used in the maintenance

task. The main role of these systems will be in the detection and diagnosis of faults. Several expert systems have been designed for diagnosis tasks in more complex environments particularly in medicine. Such systems acquire information in a question answering session with a human use and the required test results are supplied directly. The most important component of the system design is the acquisition of a knowledge base to be used by the system.

In the space station task, we envision application of expert systems in tasks requiring a knowledge base that will be easy to acquire. The knowledge base may be comparable in complexity to a repair manual for a car. The major problem in the application of expert systems on-board the Space Station, however, is that all required tests should be performed by the system. The system will have to decide the test and then solve the problem of performing the test in a given situation. Most tests will be performed using sensors; such as vision. Mobile cameras may be required so that views from different positions may be obtained. We are not aware of any existing system with such sophisticated capabilities. It appears, in fact, that expert system research has mostly neglected signal based systems applications. Some efforts have been made to design knowledge based systems for speech and vision. In general, the capability required by expert systems in space robotics will be far more demanding than the present knowledge based systems capacities.

Conventional expert systems store knowledge using production rules. These rules are very powerful for representing hard facts or facts with a confidence factor. There are several unresolved problems for the representation of signal knowledge. It is possible to use signal processing (low level vision) techniques to abstract information, and to use this information in an expert system. A major problem in this approach is the absence of abstraction techniques for visual signal under different conditions. Computer vision research in the last few years is very encouraging in this direction.

A good method of knowledge acquisition in such systems is through interaction with the on-board supervisor in a novel situation. This method of knowledge acquisition has been used in a few existing systems but the existing techniques will have to be modified to work with real-time signals.

### 3.3 Technology Forecast

Since this Space Station applications study project is to address the utilization of robotics systems to perform assembly and maintenance functions in the time frames of 1995 and 2000, we attempt to do robotics technology forecasting with the following foci:

- (a) Time horizon - 10 to 20 years
- (b) Work functions - assembly and maintenance (inspection and monitor)
- (c) Work environment - outer space.

Technology forecasting is not an exact science. However, it is possible to make an informed judgment on the characteristics of future technology on the basis of the dynamics of technological changes, expert opinions, and certain assumptions of socioeconomic and technological trends. The results presented in this section are based on the review of pertinent literature, interviews with knowledgeable persons outside the project team (see Appendix I for a list of interviewees), as well as discussions among the co-investigators, especially those who are leading relevant robotics technology research activities at the Robot Systems Division of the University of Michigan's Center for Robotics and Integrated Manufacturing (CRIM).

The first two subsections (3.3.1 and 3.3.2) of this section will present two complementary overall robotics technology forecasts relevant to this project - applying the two commonly used technology forecasting concepts of generation dynamics and precursors, respectively. The remainder of the section will discuss specific future technologies in each of the four robotic subsystems considered in this report.

### 3.3.1 Generation Dynamics

The concept of generation dynamics is based on the observation that modern technology has generally progressed in an orderly fashion, given a "surprise-free" global environment. The basic assumptions underlying such environment are [Tur 54]:

1. No drastic and unexpected technological breakthroughs
2. No major international conflicts
3. No large-scale international depression or monetary catastrophe

It has been observed that, under such an environment, quantum improvements of technology take place between generations. The successive generations of technology in a given area tend to be equally spaced over time because it takes certain time and effort to convert research and development into commercially available products in industrial dynamics, and because the leading companies in an industry need certain time to write off their investment to reap reasonable profits.

#### 3.3.1.1 Computer Generation Dynamics

A commonly cited example of generation dynamics for technology forecast is the orderly change of computer technology as follows [WCY 80]:

- 1st Generation - Vacuum tubes, 1950s
- 2nd Generation - Transistors, 1960s
- 3rd Generation - Integrated Circuits (IC), 1970s
- 4th Generation - Very large scale integrated circuits (VLSI), 1980s (present)
- 5th Generation - Artificial Intelligence (AI), 1990s

Note that the time spacing between computer generations has been about 8 to 12 years. Also, the quantum improvements may be in terms of either hardware or software. It should also be observed that, throughout the history of computer technological changes, the characteristic definition of each new generation of computers shifted over time. Moreover, there was no complete consensus at any given time on what the most prominent characteristics of the next generation of computers were going to be, and when the next generation would arrive.

### 3.3.1.2 Industrial Robot Generation Dynamics

In our literature review and expert interviews in the area of robotics, we have found a diversity of opinions regarding the generation dynamics of robots. One of the pioneers in the field has written explicitly about the characteristics and timing of three generations of industrial robots [Eng 72;Eng 80], and his views may be interpreted as follows:

1st-generation robots (1960s) - example: Unimation robots

- \*Few degrees of freedom
- \*Teach and playback facilities
- \*Small memory
- \*Relatively poor positioning accuracy
- \*Relatively low weight handling capacity
- \*Relatively low reliability

2nd-generation robots (1970s) - example: PUMA arms

- \*Six degrees of freedom for any position and orientation within the work envelope
- \*Computer control
- \*Memory options ranging from 32 to 1024 program steps
- \*Positioning accuracy repeatable to within 0.3mm
- \*Weight handling capability up to 150kg
- \*High reliability - with not less than 400 hours mean time between failures (MTBF)

3rd-generation robots (1980s) - example: GM robots with vision

- \*Over 6 degrees of freedom
- \*Microprocessor sensor processing coupled with computer control
- \*Large memory capability
- \*Adaptive and flexible control
- \*Multiple appendage hand-to-hand coordination
- \*Self-diagnostic fault tracing

### 3.3.1.3 Robot Intelligence Generation Dynamics

The time spacing between robot generations as described above, as in the case of computers, is approximately ten years. However, the above view of robot generation is not shared by all experts in the field. The difference of views appears to depend on the specialty or the subsystem with which the expert is particularly familiar with. For example, if one focuses on the

intelligence level of the robots, the following generation dynamics may be more meaningful [Nil 83]:

- 1st generation - Playback robots, late 1950s
- 2nd generation - Computer controlled robots with feedback, circa 1975
- 3rd generation - Robots with vision, circa 1983
- 4th generation - Robots programmed by advanced language, late 1980s
- 5th generation - Robots that can understand natural language, 1990s
- 6th generation - Mobile intelligent robots (can traverse streets), 2000's
- 7th generation - Robots that perform a broad range of autonomous functions, beyond 2000's

Note that the time spacing between generations from this perspective is 7 to 8 years - a more dynamic or optimistic view toward the advance of robotics technology.

By contrast, one could focus on the mechanical subsystem and derive a rather pessimistic view toward robot generation dynamics. For example, it has been pointed out that none of the robots which have been built so far have their mechanical subsystems truly responsive to intelligence [Tes83]. Thus, they are all first-generation robots even though they have been put to increasingly complex applications. In order to achieve responsiveness to intelligence, the robot system must include real-time modeling of the mechanical subsystem so that the best way among redundant responses may be chosen on the basis of real-time adaptive control. Depending on the bandwidth of the mechanical subsystem, the real-time simulation may require the speed of 1/30 second, and the hierarchy of decision making must match the structural geometry of the mechanical parts. Given the current rate of progress and R&D support for mechanical subsystems, the advent of the second generation robots with such responsiveness to intelligence is not likely to be within this century.

Similarly, one could also focus on the sensing subsystem in the consideration of robot generation dynamics [She83]. In a generic way, the first generation robots can be regarded as those early robots without sensors, first built in the 1960s, except that they may be controlled by external devices such as limit switches. The second generation robots are those with interoceptive sensors, such as the use of motor current to sense torque for the purpose of internal feedback. The third generation robots, which have just begun to emerge, are equipped with exteroceptive sensors, such as the use of external vision for control purposes. The future generations of robots will have increasingly higher levels of exteroceptive sensing capabilities, in terms of both sensors signal acquisition and processing.

Although there is a diversity of experts' views on robot generation dynamics as discussed above, their views are not incompatible. We will use all their inputs in the latter part of this section, emphasizing one perspective or another, depending on

whether it is the overall robotic system or one of its subsystems on which we will be focusing our attention.

### 3.3.2 Precursors

The concept of precursors is based on the notion that the leadership of a given technology often resides in one sector of the society, and the advances of that technology are adapted and diffused to other sectors in a systematic process with a certain time lag. The process of technology transfer and diffusion is not fixed but is embedded in the socioeconomic system of the society and is therefore subject to change as a result of the social, political, and economic forces in the society.

The classical example of using the precursor concept for technological forecasting is the time lag between military combat aircraft speed and civilian transport aircraft speed. Empirical data indicated that there was an approximately 10-year lag between the two speeds, suggesting that there was a relatively fixed time constant for the process of technological spin-off. [Ayr69] However, the relationship failed to hold beyond 1970 as the socioeconomic conditions in the U.S. underwent some fundamental changes due to the environmental movement (with a long environmentalist protest against civilian supersonic transports) and the energy crisis (which made the SST uneconomical for civilian uses).

In a number of high technology areas including computers, electronics, and communications, private-sector R&D in the U.S. is more applied and is driven more by market competition; whereas public-sector R&D is more basic and is driven more by space and defense competition [LiH77]. However, the U.S. space program went through a fundamental change after the competition of Project Apollo, which resulted in successful lunar landings. The posture of Congress and the nation toward space changed from an attitude of what can we accomplish in space to an attitude of what is the best way to allocate our scarce national resources in view of pressing problems on earth [Kor82]. NASA went from being the 4th largest industrial economic entity in the U.S. in terms of cash flow to the 48th largest in 1976 [Cri76].

Although the NASA budget has stabilized in recent years, its relative level as a percentage of GNP has been drastically reduced. The scientists who have been retained by NASA are those in the fields at the core of NASA activities - aerodynamics, space dynamics, engines, thermodynamics, etc. The fields of computers, communications and robots have not been supported at nearly as high a level as during the Apollo days. For example, the computers in the shuttle are similar to the ones used in B70 aircraft - 1970s technology that is proven and reliable. Indications are that the operational philosophy toward robots in space is quite similar to computers in space. Reliable and proven technology in robotics will be emphasized for space station applications. In terms of technological level, robots in space will be 5 to 10 years behind that of industrial robots in the next decades, barring any drastic and unexpected change of socioeconomic forces and national



policies [Gev83]. In other words, industrial robots will be precursors of robots in space, so far as the basic technology is concerned, in spite of some recent policy development hinting at the possibility of a revitalization of the space program in the uncertain future [Fru83, Bur83].

Since technology forecasts for industrial robots (IR) are available, the precursor approach suggests that we may regard IR's in the late 1980s as having the technological capabilities that robots in space will have in the time frames of 1995 and 2000. This does not mean that the space robots will be identical, or even similar in appearance, to industrial robots. However, the basic technological levels for space robots in 1995 and 2000 are likely to be similar to those for industrial robots in the late 1980s - in terms of manipulators, sensors, controls, software languages, etc. Of course, for space applications, certain characteristics will be important - e.g., lightweight, low power, compactness, temperature tolerance, etc. Also, for this project, the work functions of assembly and maintenance will be particularly important. In the following sections, we will pay special attention to these aspects as we attempt to describe the technological levels of the subsystems of the future robots in space.

### 3.3.3 Robot Structure & Control Technology Forecast

In this section we will apply the concepts of generation dynamics and precursors to one of the space robot subsystems - robot structure and control. Substantive inputs for the forecast were obtained through interviewing Prof. C. S. George Lee and by reviewing relevant literature. Specifically we try to forecast the state-of-the-art technology of industrial robots circa 1990, as they are likely to be the precursors of space robots circa 1995-2000.

Of course, we will keep in mind the difference between industrial and space robots in terms of their general requirements and therefore the different emphasis in their respective designs. For example, industrial robots are used mainly to increase productivity, and to replace human labor in certain boring work and unsafe work environments. Thus, robot speed is a central problem in industrial robots. In contrast, space robots need not be very fast but must be highly reliable and accurate because their maintenance and repair costs are extremely large compared to earth-based systems.

#### 3.3.3.1 Overall Structure and Control

By 1995-2000, the following features of space robots should become feasible at reasonable costs:

(1) The mechanical structure as well as the manipulators and end-effectors of space robots will be highly flexible in order to meet special space application requirements. For example, tentacle-like or snake-like manipulators with many more than six degrees of freedom will become more sophisticated and reliable for space applications by 1995-2000. Currently, some of these systems

are being developed for automobile industry applications, specifically by a Swedish firm named SPINE ROBOTICS which has developed a flexible design for a robot arm and wrist. The SPINE has seven axes of motion - four in the arm and three in the wrist [Spi83]. In contrast to conventional robots, it can roll its elbow in all directions, with the tool in place, offering a very high degree of freedom of access. Such highly flexible structures will be particularly useful in the generally small area available for the construction and servicing tasks on the Space Station elements.

(2) There will be a number of sensors installed on the space robot, such as force, torque, tactile and visual sensors. Visual sensors will be used mainly for parts identification and coarse control, while force sensors and tactile sensors will be used for fine control. As stated previously, speed of robot action is not an important criterion and there should be adequate time for processing sensor data to assure satisfactory performance by the space robot.

(3) Reliability of space robots is the most important problem to be considered as compared to the ground-use robots. By 1995-2000, time-proven technology for improving reliability, ranging from sturdy but lightweight materials for mechanical systems to fault-finding and self-correcting controls will be available for space robots.

(4) Industrial robots will be lighter by the use of new materials, such as plastics and composite materials, which will make robots more stiff and alleviate the dynamics design problems of robot. Such technology will be especially important, and will be available for space robots by 1995.

(5) Dynamic control of robots can be improved substantially with real-time computer simulation. However, real-time simulation can never be perfect (e.g., gear backlash will be different after each gear adjustment or replacement). Moreover, sophisticated real-time simulation requires a great deal of computer time. Nevertheless, with only modest requirement for robot speed in space, and the continuing advances being made in both real-time robot simulation technology and workpiece redesign to better match robot capabilities, we can safely forecast that some form of crude real-time simulation for space robot dynamic control will become operable by the 1990's.

(6) Passive compliance control (e.g., by wiggling the workpiece to facilitate inserting a pin into a hole) is fairly easy to accomplish. With the development and improved use of sensors, active compliance control (e.g., by the use of force sensor feedback through a computer to manipulate the pin for insertion) is being developed now and can be commercialized by 1990 for industrial robots. Thus, active compliance control is likely to become a proven technology available for space applications by 1995-2000.

(7) Multiple robot coordination is currently at a rather rudimentary state. Most commercial applications consist of

multi-robot configured in such a way that one is outside the other's work envelope. However, more sophisticated coordination has been demonstrated and is under further development. It is likely that, by 1995-2000, technology for simple coordination among multiple robots will be available for space applications.

From the point of view of control, the robot manipulators' generation dynamics are as follows:

1947: Master and slave manipulators  
1960-1970: Servo control manipulators  
Beyond 1980: Computer-controlled manipulators  
Beyond 1990: Multiprocessor-controlled multi-robot systems

On the basis of this history of manipulator generation dynamics, we may forecast that the next generation of manipulators will be computer-controlled, adaptive, self-learning, and with a moderate amount of artificial intelligence embedded in expert systems.

### 3.3.3.2 Electric Motors

The environment of a space robot imposes some unique operational constraints under which the robot system must operate. The most obvious constraints are on weight and power consumption; both must be minimized, and also the temperature extremes and reliability must be taken into account. As far as the ground use of motors is concerned, it is a mature technique, needing no special discussion in this report. The special issues concerning electric motors for space use are as follows:

- o Efficient motor utilization
- o Stable motor characteristics regardless of extreme temperature change
- o Good power to weight ratios
- o Accurate operation
- o Reliability and good controllability
- o Lightweight

At present, commercially available electric motors for industrial robots are stepper motors which are inexpensive but sometimes lose pulses, and are therefore inaccurate in operation. D.C. torque motors are often used for reliability and offer good controllability, but can be heavy and expensive. In order to meet the requirements for space use, research is in process in several areas. First, rare earth metal rotors of pancake motors now being offered hold the promise of being lightweight and providing high torque capacity. Unfortunately, the U.S. lacks rare earth elements resources, and has to import these elements at relatively high costs. It is estimated that pancake motors with REM rotors will be commercially available by 1987. Secondly, other problems associated with motors are control problems as typified by the fine control needed to pick up sensitive objects with minimum force or torque. For space robotics applications, a new class of motor controllers may be needed which are not used in current control system design. Thirdly, the problem of efficient use of motors is

important since traditional PID (Proportional, Integral, Differential) controllers make very inefficient use of the motors. Therefore, algorithms for both manipulator and end-effector control allowing more rapid, accurate, and low-energy motion is required. Lastly, even with the success of R&D on new motors for space robots, reliability may need improvement. For current robot motors the up-time is approximately 98%, and the MTBF exceeds 400 hours. However, for newly designed motors with complicated control mechanisms, the MTBF may be initially lower. It is likely that suitable d.c. motors with dedicated control systems will become commercially available within next five years, although at increased cost reflecting development amortization.

### 3.3.3.3 End Effectors and Manipulators

End-effectors should be considered together with manipulators. In robotics terminology, manipulators refer to robot shoulders and arms, while end-effectors refer to robot hand and fingers. Traditionally, manipulators in industrial robots are mechanisms with 6 degrees of freedom to position and orient a gripper and tool in any way. The end-effectors can be a customer-made tool, or a gripper, or different sorts of sensors; for example, force sensors, torque sensors, or proximity sensors. The end-effectors usually require their own means of control and power. The problems arising in the special space robotics application differing from current industrial robots will include:

- o Flexibility, which may require more than six degrees of freedom
- o Repeatability, which may approach  $\pm 0.3\text{mm}$ ;
- o Low power consumption;
- o Standardization and modularity;
- o Minimization of spatial intrusion;
- o Coordination of manipulators, especially in the case of multiple manipulators working simultaneously;
- o Dexterity and programmability for end-effectors.

One of the needs for robotics is the development of techniques that will enable two robot arms to work in consort similar to the two arms of human assembly workers. The success of multiple cooperating manipulators will depend on correct sequence programs which can be accomplished by the use of computer simulations of robot activities to detect errors. In connection with end-effectors, the system will require dexterity and programmability as well as modularity. The interface to specific sensors attached to the end-effector should also be taken into account. At present, two-arm robots are still in the experimental stage since there is no urgent need in industry to put these robots into practical operation. However, we forecast that central controlled two-arm robots will be available by the end of this

decade with improved levels of accuracy, repeatability and dexterity.

### 3.3.4 Technology Forecast for Sensors

From the robotics generation dynamics as shown in Figure 3.3.4-1, we can see that the highlight in the next stage in robotics development will be the use of sensory devices to make the robots more intelligent in perception, inspection, and other functions. What kind of information can the robots take in from the environment so that they can live in fairly unstructured environments? To what extent can we process sophisticated sensory data in real time and use them to implement dynamic control of the robots? These questions will guide our discussion in the following subsections, which will forecast future development of robot sensors based on their state-of-the-art and their historical developments.

#### 3.3.4.1 Robotic Vision Functions

The purposes of robot vision are to recognize objects and to determine location. These are termed as recognition data and orientation data. Some of the technologies needed to collect these data are:

- Visual sensors: vidicon cameras and solid-state diode arrays;
- Fiber optics, which provides certain unique robotic application opportunities;
- Computer vision processing, which includes the interpretation of multiple levels of grayscale or color in a scene with 3 dimensions;
- Direct range measurement; e.g., scanning laser radar or the use of phase or intensity measurements from LED's and photodiodes.

Note that the development of advanced and sophisticated robot vision system will probably be achieved only with the advances of all the related technologies. At present, these related technologies are only in the prototype stage. The progress of these techniques to the production stage will take time to achieve the needed reliability, complexity, and reasonable cost for practical use. The most optimistic forecast for such achievement is on the order of 8 to 10 years away. On the other hand, specialized vision with simple applications can become practical in the 1980s, and more sophisticated systems that can interpret 16 levels of grayscale in a scene will be commercially available by 1990.

The advent of sophisticated robot vision system is important, not only for the purpose of building an intelligent robot, but also for the purpose of reducing manipulator accuracy requirements by making real-time adjustment practical. We forecast

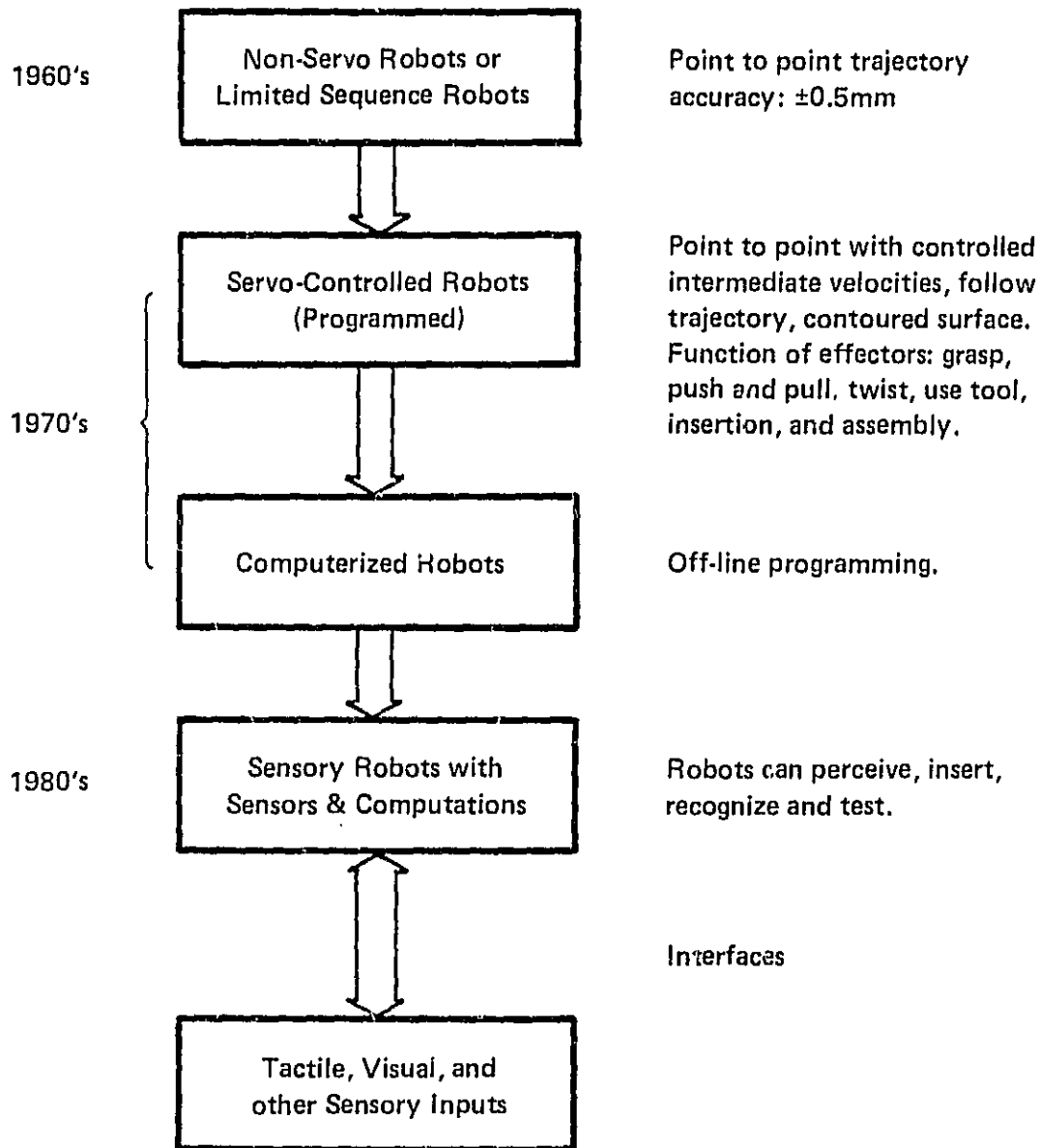


Figure 3.3.4-1 Robotics Generation Dynamics

that robots with vision will be 15% of all robots sold by 1985, 25% by 1990, and that robots with scene analysis capability will be 15% of all robots sold by 1990.

#### 3.3.4.2 Perspective on Visual Sensors

This subsection is written on the basis of an interview with Professors Edward Delp and Ramesh Jain, co-investigators of this project, and as supplemented by some literature review.

Historically the first basic visual sensor was the television camera developed before World War II, becoming commercially available around 1940. However, much was left to be desired in terms of reliability, power consumption, heat generation, accuracy, contrast, brightness, bulkiness, color-sensitivity, and longevity. The TV camera went through a great deal of improvement along with the rapid expansion of the TV industry after WWII. By 1965, partly due to the development of the picture phone, digitized video signal became commercially available, with continuing improvements in accuracy and resolution. Although new special TV cameras (with wider bandwidth, etc.) continued to hit the market, the heart of the visual sensors remained to be the vidicon tube, which has a life of about 300 hours.

It was around 1970 when solid state visual sensors emerged that featured for the first time long life, low power consumptions, and no ballooning and other problems looking into intensive light sources. Initially these were charge-injection devices (CID sensors). They are sensitive to infrared (IR) and therefore require the use of IR filters. The space resolution was relatively poor and only black and white images could be obtained. By 1976, the charge-coupled devices (CCD sensors) became available. The CCD TV camera has much higher space resolution (512 x 512 picture elements or pixels) and can take color pictures. The price of such cameras have stayed rather high because of low-yield manufacturing. However, their performance has continued to improve providing high dynamic response, lower blemish rate, and increased spectral purity.

The next generation of visual sensors is expected to include some image processing on the solid state chips in the camera. Simple analog processing is already being achieved and should become widely available by 1985. However, sophisticated processing, such as lens correction integrated into the chip, is still a decade away. The same is true for non-rectangular grids - for example, a spiral grid which is thought to be more in tune with the human-eye scanning motion.

Range sensing will make visual sensing three dimensional. The use of laser for range sensing is already available. The results are good but the speed is slow. By contrast, the stereo approach (two or more cameras) to range sensing is faster but current results are relatively poor. Improvements will be made in the future for stereo sensing through higher-power data processing. Other approaches can be used for range sensing, such as structured

lighting, i.e. projection of a known geometric pattern on the object and then processing of the image to determine range.

Other uses of visual sensors include texture sensing, which can be accomplished by analyzing the grayness variance of the image. "Purposive vision" can be obtained by deliberately moving the camera around in order to get different views of the same object. Multiple visual sensors (including the use of optical fiber links) can be mounted on the tentacle-like manipulators mentioned in section 3.3.3.1. Hence many innovative applications of visual sensors for robotics will be developed in the near future.

#### 3.3.4.3 Laser Sensors

Today's commercially available vision systems provide 2-dimensional binary pictures and operate only under carefully controlled lighting condition. In order to minimize the complexity of the resulting image, the light condition is specially controlled. The ordinary light sources are not always the best choice. In many inspection applications laser light has proved to be more effective because of its brightness, coherence, beam directionability, and monochromaticity. Another application of laser light is in rangefinding. Accurate rangefinding is required for docking and positioning with respect to other objects. Time of flight is one of the rangefinding techniques, based on the concept of sending out a signal to bounce off an object and timing its return, hence determining the distance to the object. Optical radar (laser radar) is one of the approaches, which offers the promise of greater accuracy due to higher frequencies but would require a minimum operational range below current typical designs. The potential application of laser as a range sensing element for robots needs to be further investigated.

#### 3.3.4.4 Tactile Sensors

Robots with vision and tactile sensing are of high priority in R&D. In tactile sensing, the most important quality is physical interaction data. When a part comes into contact with a workpiece, the system needs to sense the event and command the robot to act "instinctively" to complete the operation - whether it is a removal or an assembly operation. Touch sensors will increasingly eliminate the need to position workpieces precisely and will allow the robot to make logical decisions about how to adapt to varying work environments and changing tasks. The current research in tactile sensing is to develop an artificial skin which can determine position, orientation and identity of parts by touch alone. This also includes tactile arrays with conductive elastomers which are currently in the laboratory prototype stage. For industrial robots, it has been forecasted that robots with simple tactile sensors will be 5% of all robots sold by 1985 and 20% by 1990. In the recognition and control field, efforts are being made to improve pattern recognition of the objects by means of visual and tactile sensors, and to improve the adaptability of robots to the universe by using such sensors. Special development efforts should be directed to practical application of these



techniques.

#### 3.3.4.5 Force Sensors

Among non-imaging sensors, the most useful is the force sensor built into the end-effector of the manipulator. This device is able to detect a general force in the three dimensional space. The objective here is to integrate microelectronic chip technology at the sensor in order to make the associated computation in real time. This development of a smart force sensor is seen to be essential in the near term. Therefore, the development of force sensors is associated with the development of VLSI chips and the computer software to interpret the sensing data in the same manner as for tactile sensing.

#### 3.3.5 Technology Forecast for Computers & Peripherals

As far as the robot computer system is concerned, there are two types of requirements that must be considered. One is the functional requirements and the other is the operational constraints. The functional requirements of a space robot include control of all subsystems and individual experiments, machine intelligence, onboard data management and man-machine communication for supervisory control. The most obvious operational constraints are on the weight and power consumption which need to be minimized. In the following subsections we will discuss robot computer architecture, memory and other aspects.

##### 3.3.5.1 Robot Computer Architecture

The individual memories and processing units of a computer system must be integrated in order to fulfill the system requirement. From the standpoint of computer architecture, space robot computer systems should be organized into at least two (and possibly more) levels. The lower-level computers would process information from simple sensors and operate the joints, while the higher-level computer(s) decides how the various joints should move in concert to carry out a task-specific program of activities, and communicate with its operator, external computer, and other robots. Both the higher-level and the lower-level computers should meet the specific requirements and be compatible with each other. For example, the overall performance of the manipulator and the cooperative robots depends greatly on the intelligence and speed of the higher-level computers. The processing of visual and tactile sensing data by the lower-level computers must be reliable and sufficiently fast to provide timely information to the higher-level computers. At present, computers that have been qualified for space applications, such as the special requirements of temperature, radiation, vacuum, vibration, and reliability, do not use the state-of-the-art technology of earth-bound computers. For example, a state of the art space-qualified computer typically performs only 250,000 simple operations per second, and has access up to 60K words of 32 bit memory. It weighs 23 kilograms and consumes 25 watts of power. Closing the gap between qualified space computers and the state-of-the-art earth-based computers requires more development. For the reasons discussed in Section

3.3.2, we forecast that space qualified computers will continue to lag earth-bound computers by 5 to 10 years in terms of state-of-the-art technology.

### 3.3.5.2 Memory Technology

In the space robotics application, a 24- or 32-bit computer might be necessary in certain applications as 16-bit microcomputers commonly used today may not provide sufficient precision needed for manipulation, vision, tactile sensing, training, and control. High-speed computers will be increasingly used in servo calculations, image processing, force feedback, coordinate transformation, and dynamic calculations. The following Table 3.3.5.2 shows the current state of the art of memories and forecasts the trend of development through the 2000 era.

Two new technologies on the research frontier are optical memories and superconducting systems. Optical memories will not become generally available before 1985 but in the long run will offer significant advantage over bubble memories in density, cost, reliability, power consumption, and speed. Superconducting computers offer the advantage of very high speed. However, superconductivity has only been achieved below 4.2 degrees Kelvin. It is possible that this figure may increase to 35 degrees by the year 2000, and, if so, the very low temperature environment available in outer space might provide an opportunity for using superconducting systems.

### 3.3.6 Technology Forecast for Robotic Software

#### 3.3.6.1 Sensor Signal Processing

It is not clear yet how much intelligence as well as perception can be obtained in real time from interpretation of sensory information. The limits here are software and high speed computation. At present, for visual data interpretation, scenes are digitized, thresholds are formed, the data is fed to the computer, and algorithms are available for interpreting the vision scenes. Those who ultimately write programs that enable a vision module to give precise location data to a robot will have made a major contribution. The critical need is for processing large quantities of data at or near real-time rates. For example, a monochrome TV signal is represented by a data stream of approximately 250K 8-bit bytes per image and 30 images per second. This is an average data rate of 7.5 M bytes per second. To improve the speed of visual processing is a formidable task in sensor signal processing software. Widespread use of both visual and tactile sensory data processing in practical assembly work is not expected at least until 1990.

#### 3.3.6.2 Robotic Language Structures

In this subsection we will consider the generation dynamics of robot languages (RL) on the basis of literature review and discussions with Professor Kang G. Shin, a co-investigator of this project. Currently available RL's have been classified and

Table 3.3.5.2 Memory Technology Trends

| Kind of Memory/Speed   | 1982                        | 1990                        | 1995                           | 2000                           |
|------------------------|-----------------------------|-----------------------------|--------------------------------|--------------------------------|
| Semiconductor Memories | $10^5$ bits/cm <sup>2</sup> | $10^7$ bits/cm <sup>2</sup> | $10^8$ bits/cm <sup>2</sup>    | $10^9$ bits/cm <sup>2</sup>    |
| Bubble Memories        | $10^7$ bits/cm <sup>2</sup> | $10^9$ bits/cm <sup>2</sup> | $10^{10}$ bits/cm <sup>2</sup> | $10^{11}$ bits/cm <sup>2</sup> |
| Data Access Rate       | 1 Mbits/sec                 | 10 Mbits/sec                | 50 Mbits/sec                   | 100 Mbits/sec                  |
| Failure Rate           | $10^{-15}$ bit/sec          | $10^{-16}$ bit/sec          | $.5 \times 10^{-16}$ bit/sec   | $10^{-17}$ bit/sec             |

compared in terms of their actual features [BoS82]. However, a historical perspective on RL development is needed for the purpose of an RL technology forecast.

Up to around 1970, robots had been programmed only by teaching or doing. That is, human operators would move the robot arms, joints, end-effectors, etc., using teach pendants or joysticks. The robot motion is recorded and played back, with perhaps simple editorial changes through RL. Basically this type of programming is similar to numerical control with the same kind of inflexibility, for the robot would not be able to change its motion or path if something happens unexpectedly. Many industrial robots today still use this level of RL.

By 1972-73, with the spread of microprocessor applications, simple explicit languages (e.g., VAL) began to be developed for RL to specify precise motion segments. Conditional branches (if/then, goto, etc.) and simple subroutines began to be included in RL for point-to-point and primitive motions. For example, the RL can specify that if rotation exceeds 320 degrees, then the rotating motion should stop (or damage would be done to the robot). However, this generation of RL tended to be a hodgepodge of instructions added to numerical control programs, and the resultant unstructured programs were difficult to debug or to expand.

Structured programming RLs (e.g., AL) began to merge around 1976. Such languages incorporate structured control constructs, permit extensive use of coordinate transformations and frames, and allow improvements in sensors and parallel processing. In spite of these advances, RL at this level still puts more emphasis on robot motion than on the essential tasks of the robot.

The next generation of RL, which is still not commercially available, will be task-oriented. Such RLs would use high-level commands such as "Place object 1 on object 2", and would conceal low-level aids like sensors and coordinate transformations from the user. In the long run, task-oriented RL will use artificial intelligence (AI) to advantage. Such programming languages are comparatively implicit and use relatively natural language syntax. The commercialization of well-developed task-oriented RL is probably at least five years away--that is, they will become available toward 1988 or 1990.

To keep track of objects, task-oriented RL requires a world modeling system, which is a computer description of the environment surrounding the robot. Certain parts of the environment are static (e.g., the location and attributes of a table), and other parts of the environment are dynamic (e.g., the position and motion of the robot arm or objects on a moving conveyor). It is estimated that it will be two more years before commercial RLs would include at least a static world model and five years or longer before they can include a dynamic world model.

As discussed previously, generally industrial robots are going to be precursors of space robots. Thus, as far as this

project is concerned, our forecast is that only the early versions of task-oriented RLs will become available for space robot assembly and maintenance tasks by 1995 to 2000. It was suggested in Section 3.3.3.1 that tentacle-like manipulators may be particularly useful in the crowded environment in a space station. This is a very challenging area for RL development and may require several hierarchical levels of RL. Therefore, we recommend consideration of special support for structured programming RL development in this direction.

### 3.3.6.3 Robotic Artificial Intelligence

It has been said that "artificial intelligence (AI) is the technological area which needs most to be developed and mastered to accelerate robot revolution" [P1B81]. However, there is at present no real use of AI in robots today. According to Professor Ramesh Jain, a co-investigator of this project, the first use of AI in robots may be in computer vision, using a set of knowledge-based rules of thumb. However, this commercial application would be at least five years away - around 1988.

An intelligent robot has been considered [BiK81] as one capable of:

- \*receiving communication
- \*understanding its environment by the use of models
- \*formulating plans
- \*executing plans
- \*monitoring its operation

We have discussed robot communication and understanding of its environment (via world modeling) in section 3.3.6.2 on robot language. Planning and monitoring are in the realm of automatic problem solving. In the context of robotics, the problem solving will require fairly sophisticated spatial reasoning. It appears that this level of AI for robotics is at least ten years away. When two or more robots are involved, such problems as cooperative assembly and collision avoidance will require even higher level of intelligence and distributed problem solving (see section 3.2.4). Such robotics capability in a reliable and time-proven manner is unlikely to be accomplished within 15 years. It is our prognostication, therefore, that space robots in 1995-2000 are likely to use limited AI in their computer vision, but are not likely to have significant intelligence to do more than simple spatial reasoning on an individual basis. Accordingly, the use of a human supervisor in a semi-autonomous mode is the most likely implementation available in that era for Space Station tasks considered in this report.

#### 4. Support Subsystem Requirements

The previous Section 3.0 dealt with the unique robotic subsystem requirements envisioned to accomplish the multi-task mission as defined in Section 2.0. The additional subsystems to support the robotic subsystems were characterized in Section 2.4 and the general requirements for these support subsystems are delineated here. The basic functions to be performed by the support subsystems include robot mobility, command and data communications, electrical energy storage and physical latching for inertial loads transfer.

##### 4.1 Guidance and Control Subsystem

The first generation of space-station robots will have limited mobility, under close supervision, so navigation and guidance will not be major problems. This section looks beyond these early robots, to what will be required if a much greater degree of autonomous mobility is achieved.

In Section 2.4.1 it was stated that each robot must be able continuously to determine its position and orientation in the space-station frame of reference. For a robot which crawls along and is always latched to the structure this frame of reference is the only one that matters; Earth orbital parameters, orientation with respect to the Earth, and space-station rotation are unimportant. Theoretically navigation in this case could be solely by blind dead reckoning (DR) i.e. by starting from a known position and computing from distances travelled, turns taken, body rotations relative to the structure, etc., where the current position is on a stored computer model of the structure. However, in a complicated three-dimensional structure under construction, and particularly at a multi-beam joint, there would be ample opportunity for confusion and taking the wrong path. Determining orientation in such a situation by assuming that the correct structural members were being grasped would not be very reliable. Simply relying upon the robot's vision system and pattern matching capability to determine where it was in the structure might be time consuming and possibly unreliable if the same or very similar structural patterns could be seen from two or more positions. Combining DR navigation and the vision system, i.e. by predicting from the estimated position and orientation in the stored computer model what should be seen and then comparing it with the actual scene, would improve the reliability of navigation, although a multiple-beam joint might still prove difficult. Labelling every part of the space station structure with a machine-readable code would make crawling navigation a fairly straightforward matter, at the cost of complicating fabrication of structural components in orbit from basic stock. As a minimum there should be "signposts" at major structural junctions.

In the case of a robot which moves from one part of the structure to another in free flight none of the above methods of navigation are applicable. A free-flying robot must carry an inertial reference unit for short-period stabilization and guidance information. Its navigation computer must contain an accurate

real-time model of its position, orientation and motion relative to the space-station, updated at frequent intervals by reference to beacons, transponders and/or direction-and-distance devices at known positions on the structure. Probably several redundant references will be required to ensure that the robot can always obtain a position and orientation fix without risk of sight-line blockage.

The actual type or types of device to be used requires further study. Is the robot's role in determining its position and orientation to be as passive as possible, with the systems on the space station making most of the measurements and doing the computing, or is the robot to take the more active role? Radar or an optical tracker with a laser ranger can provide a position fix from a single base, but they must first acquire and maintain lock on the robot. There is always the risk of losing the target and not reacquiring sufficiently quickly, and it is difficult to service more than one robot at a time. It would seem to be more satisfactory for each robot to determine its position actively, using omni-directional ranging to at least three non-colinear transponders.

Accurate determination of the robot's real-time orientation with respect to the space station frame of reference is a more difficult problem. It requires angular measurements to be made by the robot, but to make these measurements the references on the space station must first be acquired, and the robot may be in any attitude relative to them. This implies that the tracker(s) on the robot must have spherical coverage. Since the guidance system at launch must contain accurate attitude and position information, and because the drift of the inertial system is small, the robot's computer can make a fairly good estimate at any time of where to look to find a particular beacon, the main source of error being in the computers' model of the motion of the space station. To avoid the possibility of a collision and to minimize the use of propellant, particularly on a low-speed transfer across a large structure, the guidance computer must allow for the fact that the space station may be rotating in inertial space, and, because the robot is moving relative to the space station, it is in a slightly different Earth orbit. In general, any transfer path will be curved in the space-station frame of reference. Simply pointing directly at the destination all the way, thrusting as necessary to maintain the velocity vector towards it, would be wasteful of propellant; the launch velocity vector must be computed so that the robot will coast to the vicinity of the destination without the need for further thrusting. The robot must orient itself before launch so that the thrust line for the correct transfer path is directed through its center of gravity, otherwise propellant will be lost in counteracting rotation. This of course requires that the position of the c.g. is known, allowing for any parts being carried. The approach and touch-down at the destination will require direct sensing of distance and velocity by the robot, using a radar or lidar, since its internal navigation system will probably not be accurate enough to bring it to rest relative to the local structure and close enough to be able to latch on.

The propellant used by the free-flying robot for propulsion and attitude control will may be compressed nitrogen, in a system probably similar to that developed for the Shuttle Manned Maneuvering Unit (MMU). The robot's housekeeping system must continually monitor the amount of propellant remaining to ensure that there is enough for the predicted requirements of the next transfer, plus a contingency allowance, plus the return to the recharging point.

#### 4.2 Communications Subsystems

Several technologies currently exist to provide the required communications capabilities for the robot systems. These include copper wire, fiber optics, RF, microwave, millimeter wave, optical, laser and infrared. Each of these provide different unique capabilities applicable to the various communications requirements. Moreover, the more mature technologies such as copper wire cannot be expected to provide new enhanced capabilities while the newer ones most certainly will as research progresses on them.

For intrarobot communications it is expected that wire or fiber optics techniques will predominate since the physical connection required by these methods does not impart any severe penalties with respect to weight or mobility. Wire cable in particular is a mature technology. Current serial interconnection networks can accommodate data rates of 10 megabits per second. Wire data communication paths can also be configured in parallel to increase data rates and simplify encoding/decoding requirements. Fiber optics communications paths are more immune to internal interference sources and these can provide extremely reliable communication paths.

For interrobot communications as well as robot to supervisor communications, the other techniques are more applicable. This is primarily because these other methods do not require physical connection and thus do not impose attendant mobility constraints on the robot systems.

Of these several technologies considered above, microwave is the most mature. Typical microwave communication links can handle  $10^9$  bits per second. Millimeter wave systems offer the advantage of miniaturization with attendant size, weight and power benefits.

Optical and laser systems offer almost unlimited bandwidth and provide efficient transmission of information. These systems can use free space or fiber optics. However these systems require accurate beam steering to allow the receiver to acquire the transmitted signal and thus the transmitting and receiving requirements are complicated.

#### 4.3 Energy Transfer and Storage Subsystem

There currently exists a myriad of technologies available for the transfer of energy to the robot and storage of energy within the robot. Power storage devices include primary and secondary



batteries, fuel cells, inertial flywheels and radioisotope thermoelectric generators (RTGs). Transmission methods include wire, microwave, laser and sunlight.

Primary batteries are non-rechargeable. They include lithium types, alkaline-manganese, mercury and silver-zinc. Their primary disadvantage is they must be replaced.

Secondary batteries, on the other hand, are rechargeable (within limits on the number of cycles). They include nickel-cadmium, silver-zinc, nickel-zinc and silver cadmium. Secondary batteries in general have less energy density capability than primary batteries. The primary disadvantages of secondary batteries are recharge requirements and limits on the number of recharge cycles.

Fuel cells combine hydrogen and oxygen to generate electricity. Water is produced as a byproduct. Some types of fuel cells may be recharged by electrolysis. Fuel cells have the disadvantage that the oxygen and hydrogen must be replaced (non rechargeable types).

An inertial flywheel stores energy as mechanical energy in a rotating flywheel. Primary disadvantages to this form of energy storage are low energy density and the recharge requirements.

A radioisotope thermoelectric generator (RTG) derives energy from the radioactive decay of a heavy isotope such as  $\text{Cm}^{244}$  or  $\text{Pu}^{238}$ . This type of energy storage has the disadvantages associated with the radioactivity of the fuel, particularly with regards to an accident which could contaminate the immediate area.

Transmission of energy by wire is very efficient and simple. Its primary disadvantage is the required physical connection.

Microwave power transmission is a relatively new technology that has the advantage that no physical connection is required. Its primary disadvantage is the required size of antennas for transmission and collection. As the capability to generate power at higher frequencies is developed this problem will be alleviated.

Laser power transmission is a new technology currently under active development. This technology has the primary advantage that relatively small areas are required for transmission and collection. However because of the concentration of power in the beam the risk of damage in an accident is increased.

Power can also be collected from solar radiation. This energy can be collected as thermal energy or directly converted to electricity by solar cells. A new technology currently under initial investigation is the conversion of sunlight directly to laser energy using an oxygen/iodine laser.

Space based robots will probably utilize a combination of the above technologies for power transmission and storage. The selected technologies will be chosen on the basis of tradeoffs

considering size, weight, power requirements, mobility requirements, safety, reliability, uptime requirements and accessibility.

#### 4.4 Mechanical Latching Subsystem

The latching subsystem may be considered as consisting of two primary mechanisms. The first is a set of robot latching arms which will attach the robot either directly to the space station structure or to a robot carriage which rides on a rail system. The number of latching arms must be at least three for purposes of stability, and possibly more. There is a trade-off here between the number of latching arms and the size and strength of the clamping mechanism at the end of each arm. The robot latching arms could achieve a firm attachment by utilizing an arm-end mechanism which will be a clamping device, a multi-lead short-turn screw device, or T-bar/slot device. Any of the above would be acceptable for latching the robot to the carriage. However, to latch the robot at an arbitrary location on the space station would require the clamping technique. The investigation of these and alternate latching techniques should be a topic for future studies.

The second primary mechanism could be a rail network which the robot/carriage would use to move from one work area to another as conceptually described below. A single rail type would not work under all circumstances because of space station structural variations and robot task requirements. A variety of rail types will be required, but they will have one thing in common - the robot carriage.

The robot will latch onto the carriage and also utilize a quick disconnect electronic interface such that the robot will operate and control the functioning of the carriage drive and brake devices. The carriage may be a monorail device, but its design must allow for adequate transfer of reaction loads to the space station structure.

The monorail track may be laid/attached to the space station structure by the robot and except in those areas where a permanent track is required, the robot may also detach/retrieve the rails for storage or use in another area.

It would be feasible, for construction of a large flat expanse, to lay two parallel monorail tracks with a carriage on each. A third track could be attached perpendicular to the first two and attached at each end to the two carriages. A third carriage, for the robot, would operate on the third track and would be capable of rectangular coordinate motion. The whole set-up would function in the manner of an over-head crane system. Of course electronic control of all three carriages would be maintained by the robot.

The tools and structural equipment required by the robot for the task at hand could be stored in trailers (non-driven carriages) which are hitched to the robot's carriage and pulled along with it to the work site.

In areas that cannot be reached by a robot while in a monorail carriage, it may be necessary for the robot to temporarily leave the carriage, attach directly to the space station structure, and complete the work required.

The detailed design/layout of the rail system would depend upon the design of the space station, however generic subsystems and attachment mechanisms should be investigated in research and feasibility studies.

## 5. Environmental Requirements

The Space Station robotic system will be subject to the same effects of the space environment as any other satellite or spacecraft, i.e. vacuum, thermal extremes, zero-gravity, radiation and micrometeorites. Methods of dealing with these have been developed over the last two decades or so, and they can be applied to the design of a robot and its support subsystems. However, because the consequences to the space station of a robot failure may be very serious, particularly if the robot is a free-flyer, it may be desirable to consider conservative fail-safe designs as compared with isolated satellite design features.

The most demanding problem will probably be thermal control because of the need for the robot to take up whatever orientation the current task demands. It may be difficult to deploy radiators or solar absorber panels in an optimum manner while working in close proximity to the space-station structure. Some form of active control will almost certainly be necessary, e.g. adjustable blinds over redundant radiator/absorber panels so that minimum heat loss, heat input from the sun, or radiation to space can be selected as required. Thermal control of the manipulator arms, if they need it, may be more difficult because of their larger area/volume ratio. However, the problems of motors, joint bearings, etc., operating over a wide range of temperatures in a vacuum appear to have been solved successfully in a number of Earth orbital, lunar and planetary probe vehicles, and most notably in the Shuttle manipulator arm, so there is no reason to expect unmanageable difficulties in a space station robot's arms.

Although the robots will work in nominally zero gravity, inertial forces will remain the same. A robot must not move relatively massive items, such as a structural beam, at such a rate that the robot cannot stop them before they collide uncontrollably with other structure, pull the robot off the structure, or twist off a manipulator. The requirement for preventing such occurrences is that the robot at least continually compares the linear and rotary energies it has imparted to the item with its maximum safe stopping capability in the distance-to-go.

The radiation hazard will probably not normally be important as far as the robot is concerned. Since it is assumed that the Space Station will be manned for at least part of its life it cannot be in a region of continuous high radiation, such as the Van Allen belts, and any environment which a crew member on EVA can tolerate should not bother the robot.

The micrometeorite hazard is difficult to assess since there is little information available from which to make a judgment. However, no significant damage due to this hazard has been noted in the literature reporting on the first eight Space Shuttle flights.

## 6. Continuing Research Areas Recommended for Space Application

### 6.1 Robot Structures and Control

The recommended research areas focus on the study of techniques to improve the overall performance of space robots for various possible tasks in the space station. The research should consider the performance of the following robotic subsystem areas for space station applications:

- (1) Improved manipulator design and control
- (2) Integrated control structure with sensory feedback information
- (3) Advanced control strategies utilizing sensory feedback information
- (4) Advanced sensors development.

The following subsections will elaborate on the proposed research topics.

#### 6.1.1 Improved Manipulator Design and Control

Current commercially available robots are designed kinematically to reach any given point in their specific work-volume without considering the efficiency of the control strategies that are imbedded in the control hardware/software. In order to design a kinematically efficient robot with advanced and efficient control for space station applications, there is a need to investigate the relationship between kinematical analysis and dynamical properties of a space robot through extensive computer simulation study. When designing a robot, there are several problems often faced by current robot designers. Some of the immediate questions are: (a) Since link and joint parameters play an important role in determining its sphere of influence, then how should these values be effectively chosen for the space robot application? That is, what are the effects of the links' length on computing the necessary applied torques to the joint actuators? What effects do re-arranging the prismatic/rotary joint sequence of a robot have on the Jacobian matrix computation in terms of singularity? Extensive computer simulation packages with interactive graphics capability should be utilized fully to design efficient space robots manipulators and control parameters.

#### 6.1.2 Integrated Control Structure with Sensory Feedback Information

It has been stated that current robot arm control technology suffers from the fact that the feedback gains are constants and a simple servomechanism is used to servo a nonlinear control system. Furthermore, the control structure frequently does not have any provision for processing and utilizing the sensory feedback signals for controlling the robot. A better solution is the use of a microprocessor-based attached processor (AP) which could be

designed around a complete dynamic model of the robot arm to compute all the nominal joint torque values plus the correction torques. This has the advantage of being able to change the feedback gains in the digital servo loop if the load is changing within a task cycle.

In general, the dynamic equations of a space robot arm can be expressed as:

$$D(q) \cdot \ddot{q} + H(q, \dot{q}) + D_{re}(q, \dot{q}, \ddot{q}) = \tau$$

where

$q$  is a generalized arm coordinate with commanded value  $q^d$ ,

$D(q)$  is a  $n \times n$  acceleration-related mass matrix,

$H(q, \dot{q})$  is a  $n \times 1$  Coriolis and centrifugal force vector,

$D_{re}(q, \dot{q}, \ddot{q})$  is a  $n \times 1$  inertial reaction force due to the motion of other objects in the space,

$\tau$  is a  $n \times 1$  generalized applied force/torque vector at the joints and  $n$  is the number of degrees of freedom of the robot arm.

The design of the AP would consider architectures providing solutions for this set of equations of motion. Since a robot arm is a highly nonlinear system, the controller will ultimately use some of the nonlinear terms in the equations of motion for feedback or feedforward components (i.e.  $\tau = f(D, H, D_{re}, \Delta \dot{q}, \Delta q)$ ). Thus research should be directed toward designing computational structures for computing these dynamic coefficients from the Lagrange-Euler, the Newton-Euler, and the generalized d'Alembert equations of motion in functional form. We believe that this approach of computing the dynamic coefficients (i.e.  $D, H, D_{re}$ ) will be suitable for most advanced control strategies because the AP will be designed to compute functions rather than to interface with a specific controller.

### 6.1.3 Advanced Control Strategies Utilizing Sensory Feedback Information

As stated earlier, the current approach to control system design treats each joint of the robot arm as a simple servomechanism. Such modeling is inadequate because it neglects the motion and configuration of the whole arm mechanism. Furthermore, there is little or no use of sensory feedback information for controlling the manipulator. The result is reduced servo response speed and damping, which limits the precision of the end-effector.

There are a number of important issues associated with the use of feedback to control the gross motion of an intelligent robot arm. An integral part of this research would be to investigate adaptive control strategies and identify those found most useful for space robots. These include: (1) design and implementation of

microprocessor-based resolved motion adaptive control, (2) investigation of decoupling control techniques with adaptive capability, and (3) development of nonlinear feedback control concepts.

In order to verify the performance of the proposed control strategies for space robotics applications, the need for comprehensive simulation as part of the research into various controllers design seems obvious. Computer simulation will be utilized fully in designing, developing, and testing of the control strategies for robot arms. For example, at the Univ. of Michigan Robot Systems Division, we have developed a simple and versatile wire-framed graphics display package on our VAX-11/780 computer to show the effect of control strategies on the PUMA robot arm. Structural dynamics, gear friction, location of center of mass of each link, etc., all present challenging problems in accurate modeling for actual robot arm control. The simulation of control algorithms would provide necessary performance verification before implementing them on a newly developed space robot configuration.

In the fine motion control phase using force and tactile feedback information, we recommend an effort in modeling and computer simulation of the insertion process for various contact configurations and implementing the proposed pattern analysis techniques for force recognition on robot arms equipped with wrist force sensors. The objective is to verify experimentally the equations governing the geometric force constraints and the quasi-equilibrium conditions of the insertion process in conjunction with the use of pattern analysis for recognizing these configurations. Furthermore, various guidance control algorithms for the fastener insertion process for space robot arms in the assembly mode can be simulated and studied. The proposed force feedback control technique will have an impact on the capability of future space robots in insertion process applications for structural assembly and servicing tasks.

#### 6.1.4 Advanced Sensor Research and Development

Research is proposed in the sensor area which should result in the realization of two specific devices relevant to robotics: (i) improved wrist force/pressure sensors, and (ii) tactile imaging arrays. Each of these is described briefly below.

##### (A) Improved Force/Pressure Wrist Sensors

The wrist sensor holds an important place in robotics and amounts to a single point monitor of the forces on the hand (gripper) with the ability to resolve them into both magnitude and direction. Such sensors must be rugged, reliable and have a high dynamic range. The monitoring processor must be able to separate applied forces from forces associated with the dynamics of motion.

For the implementation of the wrist sensor, the use of strain gauges is common. While the temperature drift associated with these elements can presumably be compensated by the microprocessor during known non-contact periods (dynamic

recalibration), the attachment of these elements to the bending members leads to unpredictable long-term drift and uncertain reliability. For non-silicon gauges, the output voltages (sensitivity) are also very low so that noise becomes a concern.

We suggest research to investigate improved force sensors using piezoresistive transducers or pressure sensors mounted directly on the wrist. Such devices would possess a high sensitivity and would allow improved directionality in resolving the force components. High dynamic range could be preserved along with high sensitivity by using several devices in each direction, each with a different beam/diaphragm thickness and hence sensitivity range. The goal will be to not only develop a reliable wrist sensor but also to determine the extent to which such devices can be used to successfully monitor what the gripper is doing.

## (B) Tactile Imaging Arrays for Robotics

Here we recommend the development of a tactile sensor for robotics. Such a device would provide feedback on the position of the workpiece in the gripper, its shape, and, perhaps, its texture. There is general agreement that such devices are needed but no reliable structures presently exist to perform this function.

The tactile imager would consist of a matrix of points each capable of resolving the pressure/force on that area to a level of about 1 part in 100. The outputs would be read out serially (or as several (parallel) serial channels) much as in a visible imager. Hence the tactile imager would be similar to a low resolution visible array with pressure as input.

The solid-state sensor area is a vital part of the technology needed for both space robots and industrial automation. While the area is relatively new and many sensor requirements remain to be defined, the needed solutions appear to be within the range of present technology.

## 6.2 Sensing and Machine Vision

In this section we recommend research areas for space robots with respect to vision and sensing.

### 6.2.1 Sensors

In many cases new algorithms for machine vision will be limited without new sensor developments. Two areas of sensing should be emphasized, i.e. those of visible light sensors and range sensors.

The visible light sensors mounted in TV-like cameras should be capable of higher resolution (512 x 512 pixels) and be very stable with respect to environmental changes and electrical characteristics. High resolution color CCD sensors should also be developed. The development of new solid-state sensor technology (other than CCD) will also have to be encouraged.



Range sensing using microwave or laser light needs to be further developed with respect to sensor research and new algorithms (discussed below). These non-contact sensors will allow greater flexibility in algorithm and computer hardware development. Also, fast range sensors capable of millimeter resolution from multiple views of the scene need to be developed

With the use of VLSI techniques, it will be possible to implement many of the front-end vision processing steps within the sensor chip. The placement of processing functions on the chip needs to be developed in coordination with which algorithms are performed on the chip. Candidate algorithms include arithmetic and logical window operations such as medium filtering, smoothing, and some forms of local edge detection.

### 6.2.2 Machine Vision

In the machine vision area, we recommend research to develop improved techniques for space robotic systems in five specific projects as outlined below.

#### Three-Dimensional Vision

The extraction of 3D shape information using various techniques and/or stereo camera information is recommended to provide higher autonomy robots.. Areas that need to be studied are fast algorithms, accuracy, shape extraction, and the use of the 3D information in object recognition. Integrating this information with CAM/CAD object models will be an important development. The use of 3D information for occluded parts recognition needs to be further developed.

A project currently being planned by CRIM is the use of range information and intensity information to examine the occluded parts problem. In the past most of depth information has been used with binary images. The use of range information as a feature per se needs further investigation.

#### Dynamic Scene Analysis

The use of temporal information from sequences of images should be further developed. This information can be used to aid in the inspection and assembly operations proposed for the Space Station tasks. Questions that need to be answered are: How does temporal information aid in recognition and how can one extract information about an object as it moves?

Potential projects for further study include the segmentation of moving objects in a scene, motion parameter estimation for object tracking and guidance, studies of optical flow, and characterization of the problem of motion due to the sensor, the object, or both.

#### Object Modelling

The important questions here are how does one recognize and handle an object based on an object model. Is shape information enough? Does one need to know more about the internal details of an object to recognize it or handle it? In many cases, ideas from computer graphics may be "borrowed" to address this problem.

The areas of potential investigation in object modelling research include the definition and study of object primitives and perspective views.

### Computational Complexity

The concept of real-time computer vision is in many ways difficult to achieve due to the extreme number of computations required at very high rates. The issue of algorithm complexity needs to be addressed to determine ways of minimizing the computational requirements. Fast serial and parallel algorithms need to be developed for special purpose computer architectures which will be utilized for vision.

Very basic questions should be addressed in machine vision research; for example given a serial algorithm that performs a certain vision task how does one derive a "fast" version of it? What techniques are required to map the algorithm to the computer architecture?

### Early Processing

The initial processing in any machine vision system "cleans up" the images by removing noise, correcting for lighting effects, performing image registration and the like. These algorithms are commonly called "low-level" image processing steps. In a great deal of vision research these algorithms are ignored. Research should be intensified in this area with respect to identifying robust image processing techniques. These will be candidates for algorithms to be implemented on-board the sensor chip.

Areas of potential preprocessing function research include (1) development of nonlinear smoothing operators that preserve the edge structure, (2) investigation of robust field flattening techniques and (3) utilization of texture analyses.

## 6.3 Computers, Languages and Communications

### 6.3.1 Languages

One of the critical issues in the application of robots to tasks in space (as well as ground based manufacturing tasks) is the development of a good robot programming language. Considerable additional research is needed in this area. This section outlines the major principles and problems involved.

#### 6.3.1.1 Hierarchical View of Robot Process

A robot process, a component of an integrated space automation system, can be regarded as having a hierarchical

structure. The levels of the hierarchy are cleanly divided, i.e. information processed at a particular level is not directly available to other levels of the structure. There are two paths of information flow: upward and downward. Downward moving data represents the flow of control command; a level may issue commands and coordination signals only to the level immediately below. Upward moving data comprises the flow of feedback information; thus, the feedback-based control of a level is closed in the level immediately above.

Information is abstracted as it flows upwards through the hierarchy: more physically specific information is processed in the lower levels of the hierarchy. Each level filters and transforms the data it receives producing a more abstract representation for further upward flow.

Figure 6.3.1-1 presents an example showing the levels and information flows in the hierarchy. The lowest level is a force controller for the actuator which drives a joint or an axis. The force controller generates a drive current or voltage and receives feedback regarding the motor torque. Above this is the velocity control level which specifies a desired servo rate and receives tachometric feedback for velocity stabilization.

The next level, position control, receives position feedback from a shaft angle encoder and closes the control loop for a single joint. The individual joints are then integrated into an overall structure of a single robot at the next higher level. It is at this level that the concept of a manipulator emerges from the separate individual joints providing the capabilities of coordinated joint control. Path tracking/control occurs at this level. Typically, touch and/or tactile sensors are incorporated here to prove the capabilities of intelligent path control as well as interaction with objects.

Motion planning occurs at the highest level of a single robot system which is interfaced to the next higher level responsible for task planning. Motion planning is mainly concerned with the generation of a path for a single robot and is unaware of the presence of other components in the system. This level can be regarded as the highest authority as far as a single robot is concerned. However, in order to handle unexpected events (e.g. the presence of a foreign object in the robot path), it should be able to communicate with its siblings. This implies that horizontal communications among siblings at this level are needed. If all the siblings that are within the same reachable region are working for a common task, coordination among these will be provided by the next higher level, namely the task planning level. Otherwise, they must be allowed to perform their own tasks independently of each other. Consequently, dynamic path changes have to be made thru horizontal communications or sensory feedback information without any assistance from the task planning level. Therefore, the robot process can be organized into a hierarchical structure but the structure should allow horizontal communications at certain levels.

Task planning is responsible for transforming the actual

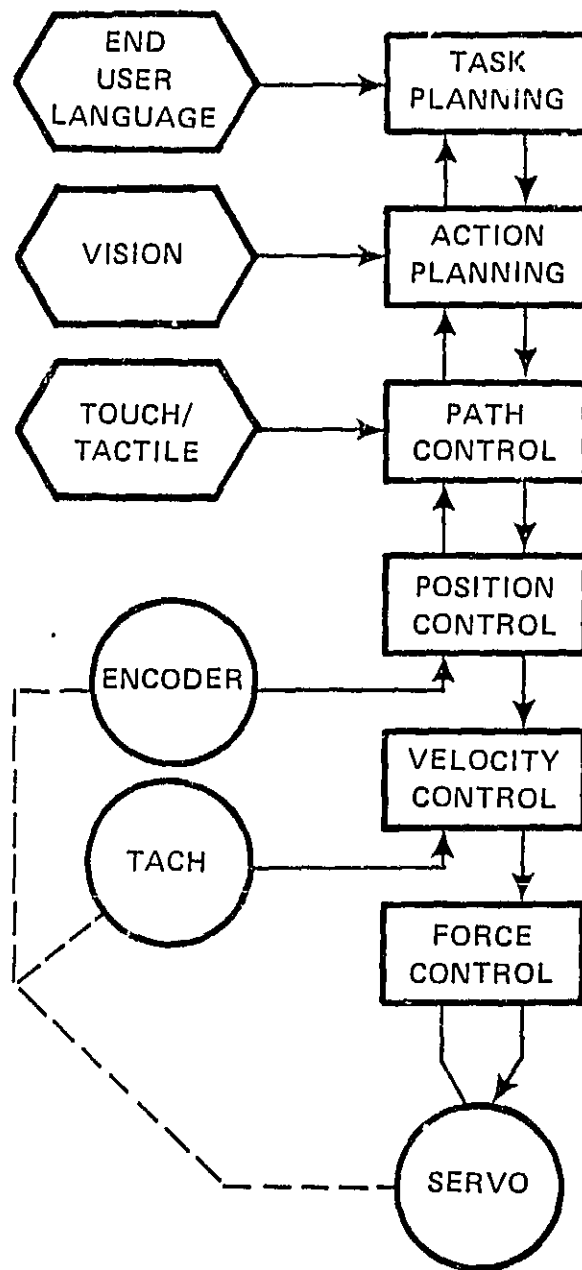


Figure 6.3.1-1 Hierarchical Structure

space robotic system application tasks (e.g. assembly, material handling or inspection) into subtasks, assigning these subtasks to individual components (e.g. robots), and then coordinating their execution by the assignees. Of course, it can assign several independent tasks to different robots; in this case it does not have to provide any coordination allowing the individuals to perform their tasks autonomously. Clearly, the latter is needed for productivity increase and is the very reason why multiple components (robots in particular) are considered.

Having decomposed the robot process into a hierarchical structure as discussed above, we have to design any RL which can run efficiently on the structure. This immediately implies a similar structure is needed for RLs. For example, a high-level synchronization command between two robots has to be interpreted and decomposed hierarchically and then assigned to the concerned levels for execution.

#### 6.3.1.2 Design of Language Features

In the development of an RL, it is essential to consider several important characteristics which apply to all programming languages in general as well as robot languages in particular.

There are a number of typical characteristics of a good programming language including:

- o Clarity, simplicity and unit of language concept
- o Clarity of program structure
- o Naturalness for the application
- o Ease of extension
- o Efficient debugging and support facilities
- o Efficiency measured by programmability and transportability
- o Portability among different hardware configurations.

Note that all of these characteristics are equally important to robot languages.

There are also additional characteristics which apply specifically to RLs but not to general programming languages in general. These include:

- o Real-time decision making capabilities
- o Interaction with external devices and sensors
- o Interactive programming capability
- o Parallel operation of multiple devices
- o Interactions with world models and CAD/CAM database systems.

In order to achieve the above desirable features, the first task of the RL design is the selection of a good set of primitives. The selection may be based on the review of the existing RLs and also anticipated future research directions. One of the most important aspects is to integrate sensory feedbacks with hierarchical organization of programs. Any RL program must

allow for the situations in which commands were not carried out as originally intended. These situations should be monitored quickly by sensors and then returned to the program. Then program must take appropriate actions to correct the situations. One of the primitives' roles is to provide the programmer with powerful but realistic tools to handle the above problem.

The most likely change in the primitives which we foresee is the increase in the semantical level of primitives. As the experience with the robotics programming grows, it is likely that new higher level primitives will be found, and they may either supplement or replace the older lower level primitives. To accommodate the change, an RL has to be designed as a hierarchical open-ended language.

The RL may consist of three largely independent groups of features:

- (i) the primitives
- (ii) data and control composition
- (iii) decomposition facilities.

The features of (ii) and (iii) can be made independent of the primitives in (i). This objective is possible to accomplish by including (a) the primitive-independent constructs, and (b) only the primitive dependent constructs that are certain to persist in RL.

The features in (iii) may be oriented towards a hierarchical organization as discussed above and comprehensive facilities for hierarchical extensibility have to be included into an RL. They may include variants of packages tailored for the specific needs of an RL. The next task of the RL design problem is to define the set of statements and data structuring capabilities which will enable a hierarchical implementation of the programs.

Also one has to include synchronization primitives in order to assure correct cooperation among parallel processes. Note that any RL must allow for parallel processing since it can provide system throughput as well as system utilization making the system more cost-effective.

#### 6.3.1.3 RL Related Subjects

An RL is expected to require very sophisticated resource environments under which the RL programs will be developed to perform the desired space application tasks. The RL environment is the collection of services available to the programmer and the programs from the operating system. Its components are libraries, editors, compilers, linkers, loaders, utilities, and the like.

The environment also includes a number of resources which must function synergistically to produce and execute robot programs. Particularly, it is important to include:

- o Debugging facilities

- o World models describing static and dynamic status of the robot environment
- o CAD/CAM databases
- o Graphics programming aids.

The effectiveness of the environment can be measured in terms of the total program development time, the on-line development time, and the level of intelligence with which robots handle real-world problems.

The total program development time is concerned with actual code generation for a given task. This may also be a function of the programmers involved including their knowledge and experience in programming. This parameter can be estimated by letting various programmers program several representative benchmark tasks. The more programmers are used the better the estimation becomes. Also, for a given set of programmers the parameters can be measured by changing the environment for sensitivity analysis.

The on-line development is related to the fact that any robot program must be verified with actual robot systems. This is a crucial part of the robot program development since, even after the verification with the world mode, it is possible to have discrepancies between the code generated and the actual action intended. The on-line verification has to be done on the real system, tying up many expensive production equipments. The amount of time for this verification may be inversely proportional to the level of intelligence and goodness of the environment. This fact implies that care must be given in the development of the environment, particularly the world model and debugging facilities.

Today's robot programming environments are primarily either simple on-line "tape recorder" modes of operation for remembering programs, or higher level text-oriented programming approaches which inherit much of the programming environment of the operating system under which they run, and use interpreters for on-line debugging. One variation of this, which is now beginning to receive wide attention is the use of graphics simulation to replace some of the on-line debugging. This is potentially valuable because (i) it may significantly reduce the on-line debugging time, thus decreasing the amount of time on expensive production equipment, and (ii) more efficient code can be generated by using compilers than interpreters.

The use of graphics aids for robot programming seems very attractive and promising. This is mainly because more powerful graphics aids are now becoming available so that one can visually simulate robot kinematics, path planning, task planning, working environment, etc. This simulation can be used to verify the code generated prior to the on-line testing. Note, however, that the graphics simulation can not handle the entire aspects of robot operation (e.g. force sensed at the time of interaction with other objects). The gain thru the simulation should be assessed by considering tradeoffs between all the factors involved: reduction in on-line debugging/development time, additional cost for

developing simulation facilities, user training cost and time, etc.

#### 6.3.1.4 Analytical Tools for Evaluating RLs

For assessment and enhancement of RL designs, it is essential to devise some tools to measure the effectiveness of an RL in terms of required attributes or features mentioned above. However, this is a difficult task. It is prohibitively expensive if not impossible to build simulators with which RLs can be evaluated. Consequently, it is logical to seek other alternatives, namely to develop analytical models which are economical and simple to evaluate. However, since the desirable RL features are highly qualitative in nature, it is difficult to make any analytical tool objective. Also, the effectiveness of an RL depends on the types of the users (e.g. English-like commands are useful to the end users but less useful to the servo control engineers).

One possible analytical approach is to first classify the users into several levels and then develop a quantitative method for measuring the effectiveness of the RL for each level of users. In this way the subjectiveness of the approach can be minimized.

Quantification of the features is in general very difficult to achieve independent of subjectivity. To alleviate this problem, scoring methods, which are popular in computer selection, can be used. These methods map different RL features into an overall effectiveness of the RL and consist of the following three steps.

- S1. Prepare a list of RL characteristics that are considered important.
- S2. For a given RL determine scores with respect to the characteristics selected in S1 and assign the relative importance of each characteristic.
- S3. Measure the effectiveness of the RL by calculating weighted sums of characteristics.

However, there are two difficult problems associated with the above approach which require further research, i.e. (1) development of techniques for assigning weights and (2) minimization of the interdependence among characteristics.

#### 6.3.2 Space Robot Computers

##### 6.3.2.1 Introduction to the Research Problem

Space robot computers will differ from their earth-based counterparts in that they will be used for extended periods in a severe space environment in an application where up-time will be critical and corrective maintenance will be difficult. Accordingly, the space robot computers will require high reliability over a long mission lifetime, fast real-time computing capability, and a need for low power consumption. Consequently, both the basic computational capacity and the machine reliability



should be addressed in the design of space robot computers.

The robot computations can be regarded as real-time computing functions which consist of data acquisition, processing, and output to actuators and displays. A real-time computer can be characterized by three important capabilities: fast response, extended I/O handling, and reliability.

Fast response will be needed for real-time applications so that the real-time system (i.e. robot system) may be responsive to the control formulated with sensor data with minimum time delay. If the delay is greater than some critical value the computed control would become obsolete and cause instability problems and/or the system would be unable to perform the required tasks. System performance will be enhanced with shorter time delays resulting from real time computational capacity.

Extended I/O handling is the essence of robot applications. The robot computer must always interact with various external devices including position and force sensors, machine vision, proximity and tactile sensors, other computers, operator(s), etc. Thus, the robot computer must have an efficient means of handling these devices (e.g. management, control, fault-tolerance of these devices).

Reliability is very important to any real-time computer since a system failure could result in a hazard to the space system and its operators. Particularly, when robots and their associated devices work cooperatively, a failure in one device will block or slow down the entire process and create a serious problem. Hence the reliability issue will be critical when the space robotics applications and their associated Space Station systems are considered.

In the following discussion we will recommend potential research directions for the design of space robot computers emphasizing the above issues.

#### 6.3.2.2 Architecture of Space Robot Computers

In order to meet the requirements of fast response, extended I/O, and reliability, multiple microprocessor (MMP) technology is a natural candidate architecture. This architecture can be justified even further by the increasing computation capability and packaging density of microprocessor and memory chips that are now becoming available. Consequently, the present discussion will be limited only to the MMP. We recommend research investigations contemplating MMP configurations for the space robotics applications including consideration of the real-time computer structure aspects.

##### A. MMP Characteristics

The MMP consists of many microprocessors, memory modules, I/O devices (even tens of thousands), and interconnection circuitry. Because of its component multiplicity, one may

immediately expect high throughput and reliability. Ideally, one may expect a linear speedup in computing power, e.g. an MMP with 10 processors may become ten times faster than a single processor computer. However, this is not the case in practice because of intercommunication delays, blocking at shared resources, etc. The speedup normally depends on many different factors such as interconnection method, applications task, task partitioning and allocation, etc.

Since the MMP has multiplicity in its components, it contains natural redundancy, thus providing high potential for fault-tolerance. However, the reliability improvement also depends on many factors including strategies for fault detection, isolation, reconfiguration, and recovery.

#### B. Real-Time Computer Structure Considerations

Typically a real-time system is characterized by sensor inputs at low baud rates; complex computations for processing these inputs; and commands to actuators and displays at low baud rates. Considering these real-time features, it is possible to discern a basic dichotomy of real-time functions from several points of view as follows:

From the point of view of data rates, there are two distinct zones; a low rate zone in which sensors report data and actuators/displays receive commands and a high rate zone in which information gathered from sensors is processed to produce appropriate commands.

Two distinct zones also exist when the point of view is the complexity of the data processing. At the sensors and at the actuators/displays, the level of data processing is low; where the computations are carried out, the data processing is at a higher level of activity.

When one considers dedication to a particular task, one again has a dual-level system. The sensors, actuators and associated equipment where the hardware is entirely dedicated to the performance of a particular set of tasks are at one level. The region where the complex data processing takes place need not be dedicated.

A natural architecture arises out of the considerations outlined above. We have a dichotomy of function, speed, and degree of processor dedication. One can choose to translate this dichotomy into concrete terms, thus creating two zones or areas: i.e. peripheral area and central area. Both of these areas require further development for the space robotics computer definition problem.

#### 6.3.2.3 Reliability of Space Robot Computers

Space robots are the typical example of systems requiring long periods of unattended operation. This fact implies that space robot computers should operate unattended for several days to

years. Note that space robot systems are less critical than the spacecraft itself, that is, they may not have to be as reliable as the spacecraft. However, in order to make the robots operational without human intervention, they have to be more reliable by far than their counterparts on the earth.

In order to provide for improved reliability and minimize down time, the space robot computers should be equipped with some level of automatic fault and error detection providing for fault location and isolation, system reconfiguration, and error recovery. Of course, 100% automation of the entire error handling would be a very difficult and expensive task. Nevertheless, efforts should be made to allow for a cost-effective approach to automatic fault/error handling in the design of space robot computers. Clearly, the four components of automatic fault/error handling must be considered together as a single package, not as separate units. For example, there are three distinct classes of error detection mechanisms, each requiring different strategies for error recovery. These three classes are signal-level detection, function-level detection, and execution of diagnostics.

#### Signal Level Detection

Usually, signal level detection is implemented by built-in self-checking circuits. Whenever an error is generated by a prescribed fault, these circuits detect the malfunction immediately even if the erroneous signal does not have any logical meaning. Typical methods of this kind employ error detection codes, duplicated complementary circuits, matcher, etc. These detection mechanisms cannot in general detect all possible errors because (i) it is prohibitively expensive to design detection mechanisms which cover all types of faults, and (ii) physical dependence between function units and detection mechanisms cannot be totally avoided. Since the signal level detects an error immediately upon occurrence, there is no contamination through error propagation. This makes the subsequent recovery operations simple and effective. Two kinds of recovery methods are suitable for this; one is error masking, the other is retry. Retry is useful to avoid errors induced by intermittent faults.

#### Function Level Detection

The function level detection is intended to check out unacceptable activities or information at a higher level than the signal level. Unlike the signal level, this level verifies system operations by functional assertions on response time, working area, provable computation results, etc. These detection mechanisms can be regarded as "barriers" or "guardian" around the normal operation. After an error is generated by a fault, the resulting abnormality may grow rapidly until it hits the barriers. Several software and hardware techniques such as capability checking, acceptance testing, invalid op-code checking, timeout and the like can be applied. Compared with the signal level detection, the function level detection is more flexible and inexpensive but error latency tends to increase.

## Diagnostics Execution

A diagnostic program supplies simulated inputs such that all existing faults are activated and thus detected. Generally the effectiveness of diagnostics is a monotonically increasing function of their run time. Since the time required for acceptable testing (close to 100%) is too long, it is not practical to apply diagnostics frequently during normal operation.

The test results only represent the system condition during the test duration rather than the system condition during task execution. Thus, no recovery action is needed even if a fault is detected. Also, a diagnostic can test every part of the system instead of just the parts commonly used for processes. There may be many units in the system which are used rarely and only under special conditions and yet have a decisive effect on system reliability; e.g. special error handling mechanisms which are activated only when a detected error is regarded to possibly cause a catastrophe. The diagnostics can usually enhance the reliability of such units at an acceptable overhead. Considering the time overhead and the properties of diagnostics, an alternative is to perform an imperfect diagnostics periodically during normal operation and perform a thorough diagnostic when the system is idle.

## Research Recommendation

Thus far, we have discussed three important error detection categories and their associated recovery methods. Future research in space robot computer reliability should include (i) the determination of an optimal combination of the foregoing three detection categories, (ii) associated error isolation and system reconfiguration, and finally (iii) the derivation of the best strategy for recovery. Note that these three are mutually dependent and therefore, they must be considered as a single entity, not separately. The final end result would be a space robot computer with automatic error handling capability.

### 6.3.2.4 Performance Measures for Space Robot Computers

The space robot system is a typical example of real-time systems. Any real-time system can be regarded as a composite of controlled subsystems (robots and other machinery for space robot systems) and controller subsystem(s) (space computers for space robot systems). Traditionally, the performance of real-time control computers has been analyzed separately from that of the corresponding controlled processes. For example, the response delay caused by the controller is neither studied rigorously nor reflected carefully into the design of control algorithms for the controlled processes. The design of control computers is frequently based on ad hoc requirements imposed by control designers. While this yields acceptable results in the control of noncritical processes, such an approach needs to be improved in the design of control computers for critical tasks e.g. space robot control. What is called for is a procedure for specifying and evaluating space robot computers, enabling systematic application

to the optimal design of space robot computer architectures and providing objective means that lend themselves to formal validation.

The use of computers as real-time controllers is becoming increasingly attractive due to continuing advances in the development of inexpensive, powerful microprocessors and memories. However, performance measures presently used to characterize real-time computers are adapted versions of those employed for more conventional computers. There is a considerable mismatch between the requirements of real-time applications and what is provided by these measures.

The performance measures discussed in the reference [KrS83] for multiprocessor controllers are designed to express the performance objectively in terms of the response time of the computer-controller. From the point of view of the controlled process (i.e. space robotic system for our case), the computer controlling it is a black box whose behavior is exemplified by its response time and reliability. It is well known that controller delay has a detrimental effect on process behavior. The new measures have to quantify a form of the time delay. We believe new performance measures including a cost function for controller response time have to be defined first. Secondly, analytical and experimental tools are needed to evaluate such improved measures. Finally, the measures must be applied to the design and evaluation of space robot computers.

Once these new performance measures are defined and estimated appropriately, they can be applied to (i) evaluation of candidate space robot computers, (ii) objective comparison of rival space robot computers, and (iii) optimal design of space robot computers. Since space robotics technology is now beginning to evolve, knowledge in the field will accrue as we gain more experience in the application of these performance measures. Consequently, a top-down approach similar to the above is essential.

### 6.3.3 Research Needs for Space Data Communications

As with the discussion on current capabilities, the areas in which future research is needed in data communication may be viewed from the perspective of topology, transmission medium and protocol. Particularly as is needed for space application these will be closely related. The requirements developed are for high speed untethered communications between any pair of processors in the system.

Current research on high speed closely spaced networks is, and is likely to remain for some time, focused on coaxial cable systems with bus or ring topologies. Both low level transmission schemes and high level protocol are under investigation. Significant results should be available by the time that space robotics designs are initiated.

The requirement for untethered communication strongly

suggests radio or light transmission in place of cable. Radio might be favored in initial development because it is less directional. As the distances involved are small, little propagation delay should be present and low powered devices should be feasible. If the low level transmission mechanisms used can be replaced while retaining compatibility with the protocol layers above them, it will then be possible to use all of upper protocol levels, say levels 2 or 3 and above in the International Standards Organization (ISO) standard. Various configurations and the transmitting and receiving capabilities to match them need to be investigated.

Even though it may be possible to replace the low level transmission capability of current (or future) networks with a radio based system, and make use of the higher levels of protocol being defined by ISO, it is not clear without further research that this is the best solution. The connection and virtual circuit establishment protocols in the ISO standards are very general and intended for dynamic connect/disconnect operation. Similarly the session establishment and file transfer protocol will be very general in nature. The space application being developed should not need that full level of generality. It is likely that some simplification and improvement in efficiency will be achievable, while still retaining the general form of the standard, by restricting the protocol to the characteristics needed by the problem at hand. This is any area which should also be investigated once the data communications requirement of the experiment have been more fully specified.

#### 6.4 Distributed Problem Solving and Knowledge Based Systems

Problem solving using multiple agents and many incomplete sources of information has received attention only recently. Many issues in these systems are being addressed by researchers. An issue that is particularly important for space robots but has been ignored so far is problem solving using sensory information in performing tasks outlined in the form of procedural knowledge for multiple robots. This problem that appears in assembly tasks to be performed by robots in space should be addressed. It appears that some issues already addressed by researchers in other environments may be pertinent to this problem. There are several novel and interesting problems, however, such as interaction of robots, that warrant attention. This problem will have applications in the factory of future on the earth also.

In problem solving using multiple robots, a representation of the work space at every point in time will be required. This representation will be 3-dimensional and will give information about the occupancy of space by objects. Since robot action planning will use this representation, careful attention should be given to the fact that this representation will be used by spatial and geometric reasoning systems. Oct-trees and generalized cylinders have been proposed for 3-dimensional space representation. Very little research has been done, however, in using these representations within a system to solve even a simple problem such as moving an object. Research in understanding the

potential uses of oct-trees and generalized cylinders for updating the model, for planning paths of objects, and for finding the precise location of objects using spatial reasoning should be undertaken. If required, new representation techniques based on solid modelling methods should be studied.

Representation of knowledge has been one of the most active fields of artificial intelligence. No good methods exist yet, however, for transformation of signals to symbolic knowledge. Some techniques have been designed for special domains. It appears that techniques from CAD/CAM for the representation of objects may be useful in space robotics, because the interest here is to work with man-made parts. A study of the suitability of this representation in inspection is required.

Production rules have been used for knowledge representation in most expert systems. Production systems do offer several very attractive features. Their efficacy in the case of an expert system working with signals is not as certain as with symbolic data. Methods to introduce iconic knowledge in expert systems may require significant extensions of conventional production systems. This research direction needs greater attention.

Most expert systems are designed to work with humans. All required tests in decision making are performed by humans and results are supplied to the system. A system for maintenance in space will have to perform all the tests that are required. This means that system should be capable of working with knowledge in the form of signals and should be able to select appropriate tests. In some tasks this knowledge may be binary and will not pose any problem. However, the information from vision modules may pose some new problems for the system. Research in the integration of testing methods using visual information will be important for the knowledge-based technician in space.

Research in spatial reasoning and planning have not received adequate attention. Spatial reasoning will play very important role in many aspects of robot problem solving in space. In many cases, it may be more useful to have a dynamic problem solving system that relies on continuous acquisition of information. This may allow approximate reasoning to be used in place of complex techniques.

## 7. Results and Conclusions

Although all future technology development predictions are subject to uncertainty, the results of this study indicate a very strong probability that selected tasks associated with the currently proposed NASA Space Station operations will ultimately be performed by some form of robot systems. The rationale for this conclusion is derived from consideration of (1) the types of on-station tasks planned by NASA, (2) the certain on-going development of earth-based robot systems with similar capability, (3) the expected realization of cost-effectivity and increased crew safety resulting from robotic task implementation and (4) the inevitable advances in component and subsystem technology which will occur and be applied to robotics technology by the 1990's time frame. Further, it is concluded that the Space Station robotic systems will be semi-autonomous, will be initially designed with a relatively small degree of on-board intelligence and will perform only simple repetitive tasks with the aid of human monitoring and intervention. As experience is gained, additional technology advancements occur and key research problems are solved, the robotic systems will become more autonomous and require less human supervision. This study does conclude, however, that fully autonomous (unsupervised) robots are not realizable for space operations in the foreseeable future.

In addition to consideration of support subsystems for mobility, guidance and control, communications, energy storage and mechanical latching, this report has emphasized the technology outlook for the critical robotic subsystems including robot structures and control, machine vision and sensors, computers and robot languages, robot data communications and knowledge based expert systems. These key subsystems have been examined here as constituent parts of a Space Station robotics applications system performing typical tasks in the construction and servicing facility modes. Following a description of the current technology status and research trends, a predicted technology forecast for the 1990's era highlighted potential research needs to achieve the required capabilities within the desired time frame.

The significant conclusion of this report is detailed in Section 6 where a series of recommendations are made for further research studies and concept development leading to verification of preliminary designs for space robotics systems. In the Robot Structures and Control area, research is recommended in manipulator design and control, feedback sensor integrated control, advanced control strategies and advanced sensors development.

To provide the required advances in Sensing and Machine Vision, further studies and investigations are suggested in the areas of visible and non-visible sensors, 3-D vision, dynamic scene analysis, object modelling, computational complexity reduction and preprocessing implementation analyses.

Augmentation and enhancement of Space Robot Computers and Robot Languages would be derived from further effort expended in the development of robot process hierarchial structures, design of



Robot Language (RL) characteristics and structures, definition of the required RL development resources and defining analytical tools for evaluating a given RL performance in the space robotic system. Further, the report concludes that near term studies should be initiated in Space Robot Computer architectures emphasizing the use of multiple microprocessors to achieve real time computational structures. A comprehensive program to define and evaluate the robotic computer reliability features and computational performance is also suggested in this report. A set of research needs for space data communications is also presented.

Finally, a recommendation is made on specific projects that should be undertaken in distributed problem solving and expert systems including development of problem solving techniques with sensor data inputs, problem solving with multi-robot systems, knowledge representation techniques, human interface considerations and research in spatial reasoning and planning.

## 8. REFERENCES

- [Alb75] Albus, J. S., "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)," Transaction of the ASME, Journal of Dynamic Systems, Measurement and Control, Vol. 97, Sep. 1974, pp. 1975, pp 220-227.
- [Arm79] W. M. Armstrong, "Recursive Solution to the Equations of Motion of an N-link Manipulator," Proceedings of the 5th World Congress, Theory of Machines, Mechanisms, Vol. 2, July 1979, pp 1343-1346.
- [Aut82a] Autovision II RAIL Reference Manual, no. MN-AV-05, Rev. 3.01, Automatix, Inc., Ed, June 1982.
- [Aut82b] RAIL Software Reference Manual: Robovision and Cybervision, no. MN-RB-07, Rev. 1.0, Automatix, Inc., Ed, Aug. 1982.
- [Ayr69] Ayres, R. V., Technological Forecasting and Long-Range Planning, McGraw-Hill, New York, 1969.
- [Bau81] E. W. Baumann, "Model Based Vision and the MCL Language," IEEE SMC Conf., pp 433-438, Oct. 1981.
- [Bej74] Bejczy, A. K., "Robot Arm Dynamics and Control," Technical Memorandum 33-669, Jet Propulsion Laboratory, Feb. 1974.
- [BeP81] A. K. Bejczy and R. P. Paul, "Simplified Robot Arm Dynamics for Control," Proceedings of the 20th IEEE Conference on Decision and Control, San Diego, Dec. 1981, pp 261-262.
- [BiK81] Birk, J. R. and Kelley, R. B., An Overview of the Basic Research Needed to Advance the State of Knowledge in Robotics, IEEE Trans. on System, Man, and Cybernetics, Vol. SMC-11, No. 8, August 1981.
- [BoP73] Bolles, R., Paul, R., "The Use of Sensory Feedback in a Programmable Assembly System," Stanford Artificial Intelligence Laboratory Memo AIM-220, Stanford University, October 1973.
- [BoS82] Bonner, S. and Shin, K. G., A Comparative Study of Robot Language, Computer, Vol. 15, No. 2, December 1982.
- [Bur83] Burbidge, E.M., Adventure into Space, Science, Vol. 221, No. 4609, 29 July 1983.
- [Cri76] Criswell, D.R., Lunar Utilization, National Aeronautics and Space Administration, Houston, Texas, 1976.
- [Der83] Deryder, L. J., Space Station Technology Systems Analysis Across Disciplines, Space Station Technology Workshop, Williamsburg, VA, March, 1983.
- [Dra77] Drake, S.K., "Using Compliance in Lieu of Sensory Feedback for Automatic Assembly," The C. S. Draper Laboratory Report

T-657, Cambridge, MA, September 1977.

- [DuD79] Dubowsky, S., DesForges, D. T., "The Application of Model Referenced Adaptive Control to Robotic Manipulators," Transaction of the ASME, Journal of Dynamic Systems, Measurement and Control, Vol. 101, Sep. 1979, pp 193-200.
- [Eng72] Engelberger, J.F., Industrial Robots--Second Generation. 1972 Symposium on Robots.
- [Eng80] Engelberger, J.F., Robotics in Practice, 1980.
- [Ern62] Ernst, H. A., "MH-1, A Computer-Operated Mechanical Hand," 1962 Spring Joint Computer Conference, San Francisco, May 1-3, AFIPS Proceedings, pp 39-51.
- [Fru83] Frutkin, A., The Politics of Space - U.S. Policy: A Drama in n Acts, IEEE Spectrum , September 1983.
- [GE82] Automation Systems A12 Assembly Robot Operator's Manual. Bridgeport, Conn: General Electric Co., Feb. 1982.
- [Gev83] Gevarter, W.B. (National Aeronautics and Space Administration), Private Communication, May 31, 1983.
- [Gro72] Groome, R. C., "Force Feedback Steering of a Teleoperator System," SM Thesis, MIT, Department of Aeronautics and Astronautics, August 1972.
- [HeC76] Hemani, H., Camana, P. C., "Nonlinear Feedback in Simple Locomotion System," IEEE Transactions on Automatic Control, Vol. AC-19, Dec. 1976, pp 855-860.
- [Hil77] Hill, J. W., "Force Controlled Assembler," Technical paper MS77-749, SME Robots II Conference, Detroit, Michigan, Nov. 1-3, 1977.
- [Hod83] Hodge, John D., Presentation to Space Station Data System Study Conference, NASA Goddard Space Flight Center, September 7, 1983.
- [Hol80] Hollerback, J. M., "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity," IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-10, No. 11, November 1980, pp 730-736.
- [HOP78] A. L. Hopkins, et. al, "FTMP-A Highly Reliable Fault-Tolerant Multiprocessor for Aircraft," Proc. IEEE , Vol. 66, No. 10, pp. 1221-1239, October, 1978.
- [IBM81] "IBM Robot System/1: AML Concepts and User's Guide," IBM Publication SC34-0411-D, September 1982.
- [Ino74] Inoue, H., "Force Feedback in Precise Assembly Tasks," MIT Artificial Intelligence Laboratory, Memo 308, MIT, August 1974.

- [KaB71] Kahn, M. E., Roth, B., "The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains," Transaction of the ASME, Journal of Dynamic Systems, Measurement, and Control, Vol. 93, Sep. 1971, pp 164-172. D. C., pp 676-679, December 1980.
- [KoG83] A. J. Koivo, T. H. Guo, "Adaptive Linear Controller for Robotic Manipulators," IEEE Transactions on Automatic Control, Vol. AC-28, No. 1, February 1983, pp 162-171.
- [Kor82] Korf, R.E., "Space Robotics", The Robotics Institute, Carnegie-Mellon University, 1982.
- [KrS83] C. M. Krishana and K. G. Shin, "Performance Measures for Multiprocessor Controllers," Performance '83, edited by A. K. Agrawala and S. K. Tripathi, pp. 229-250, May, 1983.
- [LCM82] Lee, C. S. G., Chung, M. J., Turney, J. L., Mudge, T. N., "On the Control of Mechanical Manipulators," Proceedings of the Sixth IFAC Conference in Estimation and Parameter Identification, Washington, D.C., June 1982, pp 1454-1459.
- [LeC82b] C. S. G. Lee and M. J. Chung, "An Adaptive Control Strategy for Computer-Controlled Manipulators," (Invited Paper) Proceedings of the 21st Control Decision Conference, Orlando, Florida, Dec. 8-10, 1982, pp 95-100.
- [LeC83] Lee, C. S. G., and Chen, M. H., "A Suboptimal Control Design for Mechanical Manipulators," Proceedings of the 1983 American Control Conference, June 22-24, 1983, San Francisco, California, pp 1056-1061.
- [Lee80] Lee, C. S. G., "Design Methodology of Force Control for the Sensor-Controlled Manipulators," Presented at the 101st ASME Winter Annual Meeting, Chicago, Nov. 17-20 1980.
- [Lee82] Lee, C. S. G., "Robot Arm Kinematics, Dynamics, and Control," Computer, Vol. 15, No. 12, December 1982, pp 62-80.
- [LeL83] C. S. G. Lee and B. H. Lee, "Resolved Motion Adaptive Control for Mechanical Manipulators," Proceedings of the third Yale Workshop on Applications of Adaptive Systems Theory, New Haven, Conn., June 15-17, 1983, pp 190-196.
- [Lew74] Lewis, R. A., "Autonomous Manipulation on a Robot: Summary of Manipulator Software Functions," Technical Memo 33-679, Jet Propulsion Lab., March, 1974, vol 29 January 1982.
- [LiH77] Linvill, J.G. and L.C. Hogan, "Intellectual and Economic Fuel for Electronics Revolution", Science, Vol. 195.
- [LiW77] L. I. Lieberman and M. A. Wesley, "AUTOPASS: An Automatic Programming System for Computer Controlled Mechanically Assembly," IBM J. Research and Development, Vol. 21, No. 4, pp. 321-333, July, 1977.

- [LLN83] C. S. G. Lee, B. H. Lee, and R. Nigam, "Development of Generalized D'Alembert Equations of Motion for Mechanical Manipulators," Proceedings of the 22nd Conference on Decision and Control, San Antonio, Texas, Dec. 14-16, 1983.
- [LWP80a] Luh, J. Y. S., Walker, M. W., Paul, R., "On-Line Computational Schemes for Mechanical Manipulators," Transaction of the ASME, Journal of Dynamic Systems, Measurement and Control, Vol. 102, June 1980, pp 67-76.
- [LWP80b] Luh, J. Y. S., Walker, M. W., Paul, R., "Resolved-Acceleration Control of Mechanical Manipulators," IEEE Transactions on Automatic Control, Vol. AC-25, No. 3, June 1980, pp 468-474.
- [Mar73] B. R. Markiewicz, "Analysis of the Computed Torque Drive Method and Comparison with Conventional Position Servo for a Computer-Controlled Manipulator," Technical Memo 33-601, Jet Propulsion Lab, March 1973.
- <McC68] McCarthy, J., et al., "A Computer with Hands, Eyes, and Ears," 1968 Fall Joint Computer Conference, AFIPS Proceedings, pp 329-338.
- [McD82] Manufacturing Control Language User's Manual. McDonnell Douglas Corp., 1982.
- [MuG79] S. M. Mujtaba and R. Goldman, The AL User's Manual, STAN-CS-79-718, Stanford University, Stanford, CA., January, 1979.
- [NiL82] Nigam, R., Lee, C. S. G., "A Multiprocessor-Based Controller for the Control of Mechanical Manipulators," Technical Report RSD-TR-15-82, Center for Robotics and Integrated Manufacturing, the University of Michigan, November 1982.
- [Old81] A. Oldroyd, "MCL: An Apt Approach to Robotic Mfg.," Share 56, March 1981.
- [Par81] W. T. Park, "The SRI Robot Programming System (RPS): An Executive Summary," SRI Int., March 1981.
- [PaS76] Paul, R., Shimano, B., "Compliance and Control," Proceedings of Joint Automatic Control Conference, Purdue University, July 1976.
- [Pau72] Paul, R., "Modeling, Trajectory Calculation and Servoing of a Computer Controlled Arm," Stanford Artificial Intelligence Laboratory, Memo AIM-177, Nov. 1972.
- [Pau81] Paul, R. P., Robot Manipulators: Mathematics, Programming and Control, M.I.T. Press, 1981, pp 50-84.
- [Pie68] Pieper, D. L., "The Kinematics of Manipulators Under

Computer Control," Computer Science Department, Stanford University, Artificial Intelligence Project Memo No. 72, Oct. 24, 1968.

- [PlB81] Plantier, M. and Bodmer, R., et al., Teleoperation and Automation: A Survey of European Expertise Applicable to Docking and Assembly in Space, ESTEC Contract No. 4402/80/NL/AK (SC), Geneva, EUROSTAT, S.A., May, 1981.
- [RaC81] Raibert, M. H., Craig, J. J., "Hybrid Position/Force Control of Manipulators," Transactions of ASME, Journal of Dynamic Systems, Measurement, and Control, Vol. 102, June 1981, pp 126-133.
- [Sal80] J. K. Salisbury, "Active Stiffness Control of a Manipulator in Cartesian Coordinates," Proceedings of the 19th Conference on Decision and Control, Albuquerque, NM, Dec. 10-12, 1980, Vol. 1, pp 95-100.
- [Sal78] M. Salmon, "Sigla - The Olivetti Sigma Robot Programming Language," in Proc. of the 8th Int. Symp. on Industrial Robots, Stuttgart, West Germany, pp 358-363, 1978.
- [SaL79] Saridis, G. N., Lee, C. S. G., "An Approximation Theory of Optimal Control for Trainable Manipulators," IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-9, No. 3, March 1979, pp 152-159.
- [She83] Sheridan, T., (M.I.T.), Private Communication, May 31, 1983.
- [Shi78] Shimano, B. E., "The Kinematic Design and Force Control of Computer Controlled Manipulators," Stanford Artificial Intelligence Laboratory, Memo AIM-313, March 1978.
- [SPI83] Spine Robotics Advertisement Pamphlet. Spine Robotics AB, Flojelbergsgatan 14, S-431 37 Molndal, Sweden. Phone: +46 31870710. Telex: 27955 SPINES.
- [SVM82] K. G. Shin, G. P. Vukkadala, and N. D. McKay, "Development of an Explicit Robot Control Language," Proc. COMPSAC 82, pp 282-288, Nov. 1982.
- [Tak81] Kunikatsu Takasa, "A Structural Approach to Robot Programming and Teaching," IEEE Trans. on Systems, Man & Cybernetics, vol. SMC-11, no. 4, pp 274-289, April 1981.
- [Tes83] Tesar, D. (University of Florida), Private Communication, May 23, 1983.
- [ToB62] Tomovic, R., Boni, G., "An Adaptive Artificial Hand," IRE Transactions on Automatic Control, Vol. AC-7, No. 3, pp 3-10, April 1962.
- [TML80] Turney, J. L., Mudge, T. N., Lee, C. S. G., "Equivalence of Two Formulations for Robot Arm Dynamics," SEL Report 142, ECE

Department, University of Michigan, Dec. 1980.

- [Tur74] Turn, R., Computers in the 1980's, Columbia University Press, New York, 1974.
- [Uni79] User's Guide to VAL. Unimation Inc., 1979.
- [Wa082] M. W. Walker and D. E. Orin, "Efficient Dynamic Computer Simulation of Robotic Mechanisms," Transactions of ASME, Journal of Dynamic Systems, Measurement, and Control, Vol. 104, Sept. 1982, pp 205-211.
- [WCY80] Wise, K.D., K. Chen, and R. E. Yokely, Microcomputers: A Technology Forecast and Assessment to the Year 2000, Wiley-Interscience, New York, 1980.
- [WEN78] J. H. Wenseley, et. al. "SIFT--Design and Analysis of a Fault-Tolerant Computer for Aircraft Control," Proc. IEEE, Vol. 66, No. 10, pp. 1240-1255, October, 1978.
- [Whi69] Whitney, D. E., "Resolved Motion Rate Control of Manipulators and Human Prostheses," IEEE Transactions on Man-Machine System, Vol. MMS-10, No. 2, June 1969, pp 47-53.
- [Whi72] D. E. Whitney, "The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators," Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control, Dec. 1972, pp 303-309.
- [Whi76] Whitney, D. E., "Force Feedback Control of Manipulator Fine Motion," Proceeding of Joint Automatic Control Conference, Purdue University, July 1976.
- [Wis82] Wise, K. D., ed., "Special Issue on Solid-State Sensors, Actuators, and Interface Electronics," Vol. 29, January 1982.
- [WWV82] Wolter, J., T. C. Wood and R. A. Volz, "Gripping Position for 3D Objects," Proceedings of IEEE/IAS Annual Meeting, San Francisco, CA, Oct 4-7, 1982.
- [You78] K. K. D. Young, "Controller Design for a Manipulator Using Theory of Variable Structure Systems," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-8, No. 2, Feb. 1978, pp. 101-109.

APPENDIX I. List of Interviewees for Technology Forecast

University of Michigan (UM) Experts

Delp, Edward  
Jain, Ramesh  
Lee, George  
Shin, K.G.

Non-UM Experts

Gevarter, William B.  
Nilsson, Nils  
Sheridan, Thomas  
Tesar, Delbert

NASA  
SRI International  
MIT  
University of Florida